

**UNIVERSIDAD CARLOS III DE MADRID**

**INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN  
ESPECIALIDAD EN SISTEMAS DE TELECOMUNICACIÓN**

**PROYECTO FIN DE CARRERA**



**EVALUACIÓN EXPERIMENTAL DE  
REDES MALLADAS BASADAS EN EL  
PROTOCOLO B.A.T.M.A.N.**

**Autor: María Benito Trujillo**

**Tutor: Ignacio Soto Campos**

# Índice general.

<b>RESUMEN .....</b>	<b>6</b>
<b>CAPÍTULO 1: FUNDAMENTOS DE LA REDES MALLADAS .....</b>	<b>7</b>
1.1. INTRODUCCIÓN.....	8
1.2. DEFINICIÓN DE RED MALLADA .....	9
1.3. CARACTERÍSTICAS DE LAS REDES MALLADAS .....	10
1.4. INFRAESTRUCTURA DE LA RED MALLADA .....	11
1.5. REDES MALLADAS DE PRIMERA, SEGUNDA Y TERCERA GENERACIÓN .....	12
1.6. FUTURO DE LAS REDES MALLAS .....	13
<b>CAPÍTULO 2: APLICACIONES DE LAS REDES MALLADAS .....</b>	<b>15</b>
2.1. INTRODUCCIÓN .....	16
2.2. RED DOMÉSTICA DE BANDA ANCHA .....	17
2.3. RED MALLADA DE UNA COMUNIDAD DE VECINOS .....	18
2.4. REDES DE EMPRESAS .....	19
2.5. REDES DE AREA METROPOLITANA .....	20
2.6. SISTEMAS DE TRANSPORTES PÚBLICOS .....	21
2.7. SISTEMAS EN SALUD Y MEDICINA .....	21
2.8. AUTOMATIZACIÓN DE EDIFICIOS .....	21
2.9. SISTEMAS DE VIGILANCIA .....	22
2.10. OTRAS APLICACIONES .....	23
<b>CAPÍTULO 3: PROTOCOLOS DE ENCAMINAMIENTO .....</b>	<b>24</b>
3.1. INTRODUCCIÓN .....	25
3.2. MODIFICACIÓN DE LOS PROTOCOLOS DE ENCAMINAMIENTO DE LAS REDES MÓVILES AD HOC .....	26
3.3. DISEÑO DE NUEVOS PROTOCOLOS DE ENCAMINAMIENTO.....	27
3.4. CRITERIOS PARA LA CLASIFICACIÓN DE PROTOCOLOS .....	29
3.5. EJEMPLOS DE PROTOCOLOS DE ENCAMINAMIENTO PARA LAS REDES MALLADAS INALÁMBRICAS.....	31
3.5.1. PROTOCOLOS DE ENCAMINAMIENTO PROACTIVOS: BATMAN.....	31
3.5.2. PROTOCOLOS DE ENCAMINAMIENTO PROACTIVOS: OLSR .....	34
3.5.3. PROTOCOLOS DE ENCAMINAMIENTO PROACTIVOS: DSDV.....	36
3.5.4. PROTOCOLOS DE ENCAMINAMIENTO PROACTIVOS: OSPF.....	37
3.5.5. PROTOCOLOS DE ENCAMINAMIENTO PROACTIVOS: PWRP .....	38
3.5.6. PROTOCOLOS DE ENCAMINAMIENTO PROACTIVOS: MMRP .....	38
3.5.7. PROTOCOLOS DE ENCAMINAMIENTO REACTIVOS: AODV.....	38
3.5.8. PROTOCOLOS DE ENCAMINAMIENTO REACTIVOS: DSR .....	39
3.5.9. PROTOCOLOS DE ENCAMINAMIENTO REACTIVOS: TORA .....	40
3.5.10. GPSR (Greedy Perimeter Stateless Routing) .....	41
3.6. CONCLUSIÓN DE LOS PROTOCOLOS DE ENCAMINAMIENTO .....	41

<b>CAPÍTULO 4: DESPLIEGUE DE UNA RED MALLADA BASADA EN EL PROTOCOLO B.A.T.M.A.N.</b>	<b>43</b>
4.1. INTRODUCCIÓN	44
4.2. PREPARACIÓN PARA EL DESPLIEGUE DE UNA RED MALLADA BASADA EN B.A.T.M.A.N.	45
4.2.1. CONFIGURACIÓN DE ROUTERS PASO A PASO	46
4.3. DESPLIEGUE DE UNA RED MALLADA BASADA EN B.A.T.M.A.N.	48
4.3.1. CONFIGURAMOS LA GATEWAY	49
4.3.2. CONFIGURAMOS LOS CLIENTES	49
4.4.3. COMPROBACIONES Y MODIFICACIONES	50
4.4. PRUEBAS DE FUNCIONAMIENTO	52
4.4.1. RETARDO CUANDO ARRANCA LA RED	52
4.4.2. RETARDO CUANDO CAE UN ENLACE EN UN SENTIDO	54
4.4.3. RETARDO CUANDO CAE UN NODO	58
4.4.4. RETARDO CUANDO INTRODUCIMOS UN NODO NUEVO	60
4.4.5. ANÁLISIS DE TRÁFICO DIRIGIDO A INTERNET	63
4.5. VALORACIÓN DE LOS RESULTADOS	69
<b>CONCLUSIONES</b>	<b>71</b>

## Anexo y Apéndices.

<b>ANEXO 1: PRESUPUESTO</b>	<b>73</b>
<b>APÉNDICE 1: CONFIGURACIÓN DE LOS ROUTERS</b>	<b>75</b>
PASO 1: Configuración de la tarjeta eth1	76
PASO 2: Conexión del router	76
PASO 3: Acceso al router	76
PASO 4: Cambio de password	77
PASO 5: Cambio de firmware	77
PASO 6: Recuperar el router	79
PASO 7: Cambio de IP	79
PASO 8: Acceso al router sin petición de password	80
PASO 9: Instalación de B.A.T.M.A.N. en los routers	81
PASO 10: Configuración de los parámetros de B.A.T.M.A.N.	82
PASO 11: Instalación de paquetes adicionales	84
<b>APÉNDICE 2: SCRIPTS Y RESULTADOS</b>	<b>85</b>
A2.1. SCRIPT CONEXIÓN A INTERNET	86

A2.2. SCRIPT COMPROBACIÓN DE VECINOS .....	89
A2.3. COMPROBAR EL FUNCIONAMIENTO DE LA RED .....	91
A2.4. COMPROBAR EL ACCESO A INTERNET .....	93
A2.5. MEDIDAS DEL RETARDO CUANDO ARRANCA LA RED .....	97
A2.6. MEDIDAS DEL RETARDO CUANDO CAE UN ENLACE .....	113
A2.7. MEDIDAS DEL RETARDO CUANDO CAE UN NODO .....	141
A2.8. MEDIDAS DEL RETARDO CUANDO INTRODUCIMOS UN NODO NUEVO .....	151
A2.9. CAPTURA DE TRÁFICO DENTRO DE LA RED MALLADA .....	165
A2.9.1. CAPTURA DE TRÁFICO EN EL PUERTO 4306 .....	165
A2.9.2. CAPTURA DE TRÁFICO SIN ESPECIFICAR EL PUERTO .....	170
A2.10. CAPTURA DE TRÁFICO FUERA DE LA RED MALLADA .....	183

## Índice de figuras.

Figura 1.1: Conexión total y parcial de la red mallada .....	9
Figura 1.2: Red de infraestructura .....	11
Figura 1.3: Red de clientes mallados .....	12
Figura 1.4: Red híbrida .....	12
Figura 1.5: Red de primera, segunda y tercera generación .....	13
Figura 2.1: Red doméstica de banda ancha .....	17
Figura 2.2: Red mallada de una comunidad de vecinos .....	18
Figura 2.3: Red mallada de empresas .....	19
Figura 2.4: Redes de área metropolitana .....	20
Figura 2.5: Sistemas de transportes públicos .....	21
Figura 2.6: Automatización de edificios .....	22
Figura 4.1: Distribución de los routers .....	44
Figura 4.2: Router WRT54GL .....	45
Figura 4.3: Red mallada inalámbrica con 9 routers .....	48
Figura 4.4: Caída del enlace 10.10.0.8 – 10.10.0.10 .....	55
Figura 4.5: Caída del nodo 10.10.0.10 .....	58
Figura 4.6: Red mallada de 8 routers .....	61
Figura 4.7: Datagrama IP .....	64

Figura 4.8: Cabecera UDP .....	64
Figura 4.9: Acceso a Google.es .....	65
Figura 4.10: Comparación de datagramas IP .....	67
Figura A1.1: Cambio de password .....	77
Figura A1.2: Cambio de firmware <a href="#">openwrt-wrt54g-squashfs.bin</a> .....	78
Figura A1.3: Cambio de firmware <a href="#">openwrt-brcm-2.4-squashfs.trx</a> .....	78
Figura A1.4: Reset router Linksys WRT54GL .....	79
Figura A1.5: Cambio de IP .....	81
Figura A1.6: Configuración de parámetros .....	83

## Índice de tablas.

Tabla 4.1: Tabla de nodos vecinos .....	48
Tabla 4.2: Retardos cuando arranca la red .....	54
Tabla 4.3: Retardos cuando cae un enlace .....	57
Tabla 4.4: Retardos cuando cae un nodo .....	59
Tabla 4.5: Retardos cuando introducimos un nodo nuevo .....	62
Tabla 4.6: Retardos en las distintas situaciones .....	69

## Resumen.

Las redes malladas inalámbricas son una tecnología que nos permite interconectar nodos sin cables. Cada nodo que forma parte de la red mallada inalámbrica puede comunicarse con cualquier otro nodo de la red, aunque no este dentro de su área de cobertura. Esto es así porque la información viaja de nodo a nodo hasta alcanzar el destino. Los nodos que forman la red pueden actuar como routers mallados o como clientes mallados.

Las funcionalidades de los routers mallados permiten la integración de las redes malladas inalámbricas con otras redes existentes como WI-FI, WiMAX, WiMedia,... Lo que permite a los usuarios de estas redes poder acceder a otros servicios.

Las ventajas que presenta frente a otras redes son el bajo coste al utilizar enlaces inalámbricos, la facilidad de aumentar el área de cobertura incluyendo nuevos nodos, la robustez que presenta ante fallos al disponer de rutas alternativas y la capacidad de transmisión que permiten aplicaciones a los usuarios en tiempo real de voz, video y datos.

Los dispositivos electrónicos habituales, como teléfonos y portátiles, pueden conectarse a la red mallada inalámbrica, si equipamos a estos dispositivos con una tarjeta de red inalámbrica que tenga acceso a un router mallado. Otra opción es unir mediante cable Ethernet al dispositivo y al router mallado. Gracias a esta característica es posible que todos los usuarios estén conectados en cualquier lugar y momento.

Las redes malladas inalámbricas ofrecen servicios inalámbricos para una gran variedad de aplicaciones en el área personal, local y metropolitana. Estudiaremos las aplicaciones en entornos de edificios, empresas, transportes públicos y muchos más.

Aunque las redes malladas inalámbricas han avanzado, todavía queda mucho por investigar en las capas de los protocolos de encaminamiento. En este proyecto vamos a describir los protocolos de encaminamiento existentes y su objetivo. Además de las características necesarias que deben presentar los protocolos de encaminamiento para que funcionen correctamente en las redes malladas inalámbricas.

Estudiaremos la posibilidad de utilizar protocolos de encaminamiento ya existentes para las redes malladas inalámbricas y nuevas estrategias como basarse en la localización.

También incluimos un despliegue realizado en el laboratorio de una red mallada inalámbrica experimentando con el protocolo de encaminamiento B.A.T.M.A.N.. Explicando paso a paso el montaje de la red, la configuración del protocolo B.A.T.M.A.N. y las pruebas de funcionamiento. Además haremos una demostración de cómo es posible el acceso a Internet de toda la red mallada inalámbrica cuando sólo hay un nodo conectado a Internet.

# **CAPITULO 1: FUNDAMENTOS DE LAS REDES MALLADAS**

## 1.1. INTRODUCCIÓN

En este capítulo vamos a definir lo que entendemos por red mallada inalámbrica y los dos tipos de nodos que se utilizan, routers y clientes mallados.

Explicaremos detalladamente las características que presentan las redes malladas inalámbricas, como la robustez, la topología dinámica, el ancho de banda limitado, la seguridad y los canales de comunicación aleatorios.

Además expondremos las diferentes redes que existen dependiendo de la función que desempeñen los nodos de la red mallada. Podemos encontrarnos una red donde la infraestructura esta formada por routers mallados, clientes mallados o una combinación de ambos.

Si nos fijamos en el modo que se interconectan los clientes y los routers nos podemos encontrar con 3 generaciones diferentes. En la primera generación los clientes y los routers utilizan el mismo canal de radio. En la segunda generación se utilizan dos radios diferentes, un canal de radio para los clientes y otro canal de radio para los routers. Y por último nos encontramos con la tercera generación que utiliza dos canales de radio para los routers.

Las redes malladas inalámbricas presentan ciertas ventajas que las convierten en la mejor opción en múltiples ocasiones. Una de las ventajas es que no necesita cableado para interconectar a los nodos y por tanto el coste es más bajo. Además como las redes van aumentando día a día otra ventaja que presenta es la facilidad de incluir nuevos nodos para aumentar las zonas de cobertura. Las redes malladas inalámbricas ofrecen fiabilidad en la comunicación gracias a los caminos redundante que se pueden utilizar en caso de fallo. Otro punto a favor es que las redes malladas inalámbricas son compatibles con otro tipo de redes que ya están desplegadas.

Por todas estas ventajas y todas las ventajas que se pueden conseguir estudiando y obteniendo experiencia, las redes malladas inalámbricas tienen un alto potencial en el mundo de las telecomunicaciones.



## 1.2. DEFINICIÓN DE RED MALLADA

Se denomina red mallada (mesh, en inglés) a aquella donde existen al menos 2 caminos a cada nodo. [12]

Por tanto, podemos decir que la red mallada inalámbrica es una red de nodos unidos todos con todos, o la gran mayoría, y que se conectan sin cables.

En estas redes existen dos tipos de nodos, los routers mallados y los clientes mallados. Los routers mallados se encargan de realizar funciones de reenvío de tráfico de otros nodos y funciones de gateway para interconectar con otras redes existentes. Si comparamos un router mallado con un router inalámbrico convencional observamos que se alcanza la misma cobertura utilizando una potencia de transmisión menor, ya que puede dar servicio a nodos fuera de su cobertura radio a través de otros nodos.

Un aspecto importante del funcionamiento de las redes malladas inalámbricas comparándolo con las redes tradicionales WI-FI es que no necesitan que los nodos a los que quiere transmitir estén dentro del rango de cobertura del nodo. Es decir, que un nodo puede transmitir a otro nodo que no esté en su mismo rango de cobertura, ya que se realiza un encaminamiento multi-salto y la información va atravesando nodos intermedios hasta que alcanza el destino.

Los clientes mallados también realizan la función de reenvío de tráfico de otros nodos pero no realizan las funciones de gateway. Para acceder a otras redes, como por ejemplo Internet, los clientes mallados deben conectarse a los routers mallados que permiten el acceso a otras redes gracias a la función de gateway.

Como ya hemos comentado las redes malladas inalámbricas pueden tener una conexión total de todos sus nodos o parcial. Si la conexión es total la red es muy robusta ante fallos en la topología, pero resulta muy costoso y por este motivo se utilizan la alternativa de conexión parcial que también presenta una gran robustez.

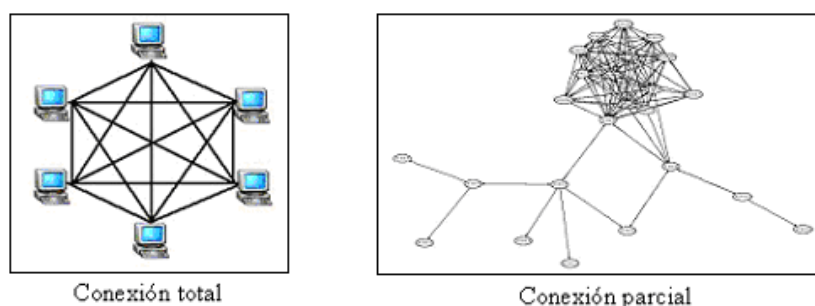


Figura 1.1: Conexión total y parcial de la red mallada [3]

El principal objetivo de las redes de comunicaciones es que la información llegue a su destino de manera rápida y fiable. En este aspecto las redes malladas inalámbricas

ofrecen un buen servicio porque son robustas ya que poseen redundancia de enlaces y de nodos en caso de fallo. Cada nodo de la red mallada posee al menos dos conexiones a dos nodos diferentes, por lo que si se cae un enlace siempre se puede alcanzar al nodo por otro enlace.

Las redes malladas al ser inalámbricas presenta ventajas a la hora de incrementar el número de nodos, ya que no es necesario cambiar infraestructuras como en el caso de las redes cableadas. Y por tanto se puede incluir un nuevo nodo en cualquier momento y lugar. Como consecuencia el coste de este tipo de redes inalámbricas es mucho menor que en las redes cableadas, ya que no hay que invertir en materiales de cableado y en estudios enfocados a la unión más óptima de los nodos.

Como en todas las redes debemos utilizar los protocolos de encaminamiento que se encargan de la transmisión de la información desde el nodo origen al nodo destino. Para conseguirlo, los protocolos de encaminamiento deben conocer el estado de la topología de la red y saber cuando se ha introducido un nuevo nodo, cuando ha fallado un nodo, cuando se satura un enlace y cuando es necesario balancear la carga por otros caminos porque hay saturación.

### **1.3. CARACTERÍSTICAS DE LAS REDES MALLADAS**

Las redes malladas inalámbricas presentan ciertas ventajas como la robustez y la flexibilidad, y otras muchas más que hacen de este tipo de red una opción muy ventajosa. Pero también existen puntos que hay que mejorar como la seguridad, el ancho de banda y los canales de comunicación. Vamos a explicar con más detalle cada una de las características de la red mallada.

En una red mallada un conjunto de nodos se comunican entre sí de manera directa transmitiendo la información de nodo a nodo hasta que llega a su destino final. La información atraviesa múltiples saltos y no hay necesidad de una unidad centralizada que controle el modo de transmisión. La comunicación se realiza entre los nodos directamente. Cada nodo puede ser origen y destino de los datos o encaminar la información de otros nodos.

Las redes malladas inalámbricas son robustas al tener varios caminos disponibles entre el nodo origen y el destino. De modo que el servicio no se verá afectado por la caída de un nodo o por la ruptura de un enlace.

Otro aspecto importante de las redes malladas es la flexibilidad, ya que se pueden incluir nuevos nodos sin implicar un replanteamiento de la red global, ya que las redes malladas inalámbricas afrontan los cambios en la topología. Además se pueden incluir nuevos nodos en cualquier lugar siempre que alcance a algún vecino que ya pertenezca a la red.

El ancho de banda es limitado ya que los nodos deben estar en contacto y realizan el transporte de datos de otros nodos y se producen congestiones en los enlaces.

En este tipo de redes donde la información atraviesa diversos nodos, se compromete la seguridad y se puede dar la situación de que un usuario que no es el destino pueda acceder a información privada de otros usuarios. Por este motivo se define una subcapa de seguridad para proteger la información.

A pesar de las desventajas presentadas, son más importantes las ventajas que ofrecen las redes malladas inalámbricas que merece la pena invertir en mejorar estos puntos.

#### 1.4. INFRAESTRUCTURA DE LAS REDES MALLADAS

Existen tres tipos de redes malladas inalámbricas si nos fijamos en el funcionamiento de los nodos. Como hemos explicado, las redes malladas tienen dos tipos de nodos, los routers y los clientes mallados. Por tanto nos podemos encontrar una red donde la funcionalidad de la red reside únicamente en los routers, en los clientes o una combinación de ambos nodos. [4]

- Red de infraestructura: En este tipo de red los routers mallados forman la infraestructura principal de la red y se conectan entre ellos de manera inalámbrica. Permiten el acceso a clientes mallados y a clientes convencionales de otras redes. Los clientes convencionales se pueden conectar a los routers mallados a través de cable Ethernet o de manera inalámbrica.

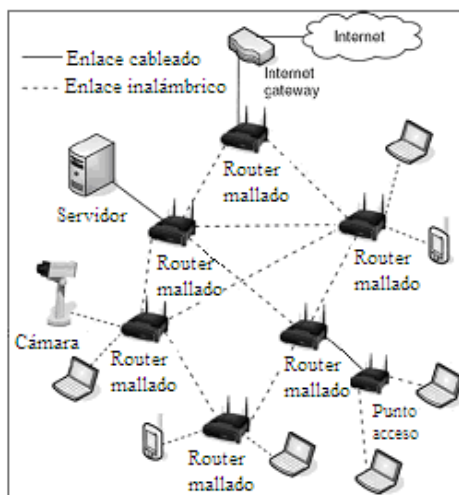


Figura 1.2: Red de infraestructura [4]

- Red de clientes mallados: En las redes de clientes mallados no se utilizan los routers mallados. Los clientes mallados forman la infraestructura de la red y se conectan de manera inalámbrica. En este caso los clientes mallados actúan como cliente y router para el resto de nodos.

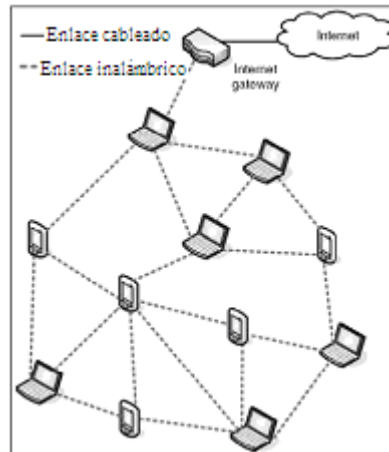


Figura 1.3: Red de clientes mallados [4]

- Red híbrida: Como su propio nombre indica es una red que combina los dos conceptos anteriores. En las redes híbridas los routers y los clientes realizan las funciones de encaminamiento.

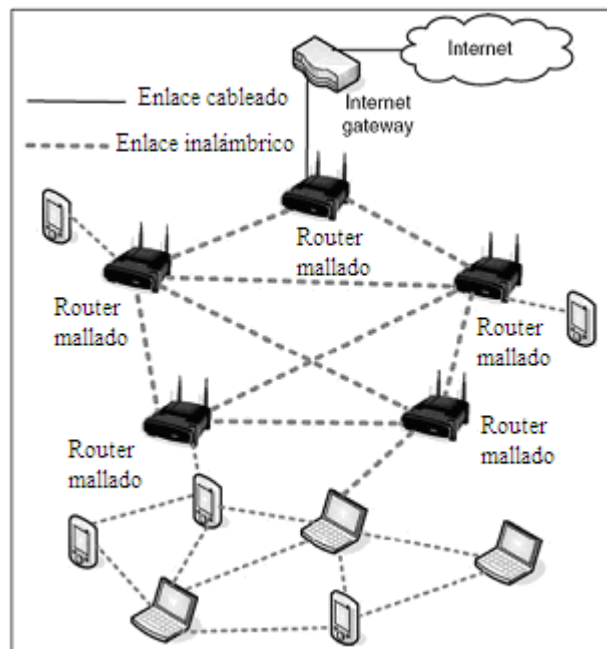


Figura 1.4: Red híbrida [4]

## 1.5. REDES MALLADAS DE PRIMERA, SEGUNDA Y TERCERA GENERACIÓN

Existen 3 generaciones de redes malladas basándonos en el modo en el que se usan los canales radios disponibles para interconectar los nodos. [7]

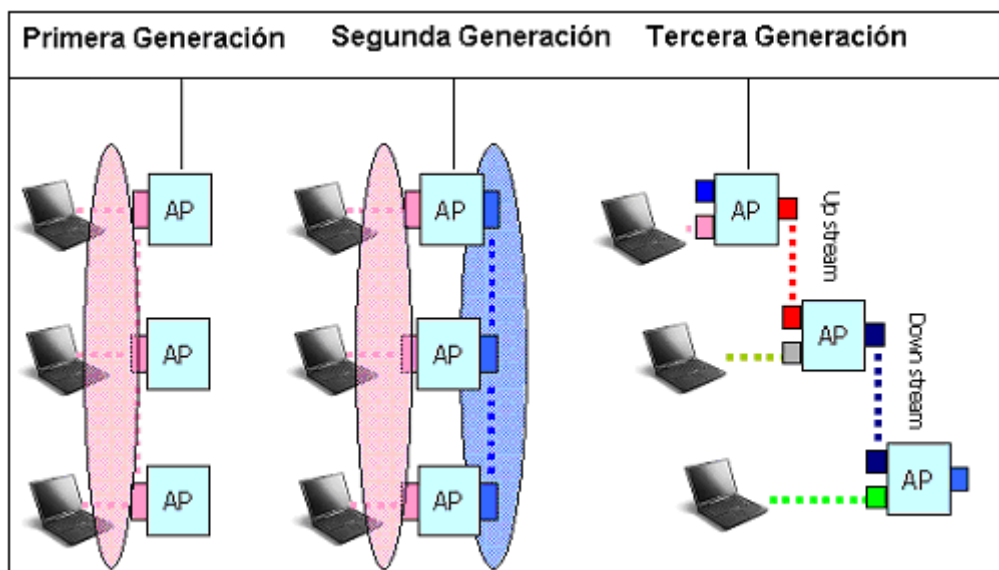


Figura 1.5: Red de primera, segunda y tercera generación [4]

En las redes malladas de primera generación los clientes y los routers comparten el mismo canal de radio. Como es obvio si un cliente o router está transmitiendo el resto no pueden transmitir ya que se produciría colisión. El ancho de banda de esta situación se reduce considerablemente cuando el número de routers aumenta. Cuando un router está transmitiendo no puede recibir y se reduce el ancho de banda a la mitad. El ancho de banda disponible se puede calcular en función del número de saltos, es decir, si en cada salto perdemos la mitad del ancho de banda, entonces el ancho de banda total disponible para el enlace Ethernet es  $1/2^N$  (por ejemplo si tenemos un número de saltos igual a 5,  $N=5$ , el ancho de banda que nos queda para un usuario es  $\frac{1}{2} * \frac{1}{2} * \frac{1}{2} * \frac{1}{2} * \frac{1}{2} = 1/32$ ).

En las redes malladas de segunda generación se solucionan los problemas de colisión de la primera generación porque se disponen de dos canales de radio diferentes, un canal de radio para la conexión de los clientes y otro canal radio para la comunicación entre los routers (backhaul). Aunque se solucionan los problemas de la primera generación todavía existen inconvenientes cuando hay alta demanda de uno de los routers ya que provoca congestión en el backhaul.

Por último tenemos las redes malladas de tercera generación que solucionan los problemas de las dos generaciones anteriores. Utiliza dos canales de radio para el backhaul, uno para comunicarse con el router raíz (upstream) y otro para comunicarse con el router vecino (downstream). Esta opción es la que mejor funcionamiento presenta porque los routers pueden comunicarse con sus vecinos sin tener que depender de si el resto de routers esta transmitiendo o recibiendo.

## 1.6. FUTURO DE LAS REDES MALLADAS

Las redes malladas inalámbricas presentan un alto potencial para los futuros despliegues de redes debido a sus ventajas. Por un lado tenemos que las redes malladas inalámbricas ofrecen una manera sencilla de desplegar una red sin necesidad de

cableado. Además presentan costes y consumos bajos, que son puntos a tener en cuenta cuando hay que realizar un estudio de implementación de una nueva red.

Por otro lado tenemos que las redes malladas inalámbricas son robustas respecto a que la información no se pierde ante cambios topológicos en la red y por este motivo es interesante estudiar este tipo de redes que aseguran la transmisión satisfactoria. Por ejemplo en amplias zonas de exteriores son robustas gracias a la redundancia y recuperación ante fallos de manera automática.

Otro punto a favor de las redes malladas inalámbricas es que se complementa con otras redes existentes, como las redes metropolitanas de fibra óptica o la tecnología WiFi que está presente en una gran variedad de dispositivos que todo el mundo utiliza hoy en día, como son los teléfonos, portátiles, PDAs,...

Además como las redes van creciendo y se necesitan cubrir más zonas, es importante el uso de redes malladas inalámbricas ya que permite la inserción de nuevos nodos de manera sencilla.

Y no solo se utilizan las redes malladas para el área de las telecomunicaciones sino que también se utilizan en otras áreas como sanidad, formación, turismo, cultura y ocio.

Por todos estos motivos las redes malladas inalámbricas presentan suficientes motivos para seguir investigando y mejorando.

## **CAPITULO 2: APLICACIONES DE LAS REDES MALLADAS**

## 2.1. INTRODUCCIÓN

El desarrollo y la investigación de las redes malladas inalámbricas esta motivado por aplicaciones prometedoras que otras redes inalámbricas o cableadas no pueden soportar. Algunas de estas aplicaciones son las redes domésticas de banda ancha, redes comunitarias y redes metropolitanas.

Las redes malladas inalámbricas pueden cumplir con los requisitos de las redes domésticas de banda ancha si incluimos un router mallado para interconectar a los diferentes dispositivos y ofrecer el acceso a Internet.

En las redes comunitarias el uso de redes malladas proporciona ciertas ventajas como que la información entre los vecinos no viaja por Internet y que se cubre una amplia zona de cobertura.

Al utilizar redes malladas inalámbricas en las redes de empresas se mejora la robustez y ofrece la posibilidad de compartir recursos de la empresa gracias a las funcionalidades de los routers mallados.

Y una aplicación mucho más ambiciosa es utilizar las redes malladas inalámbricas en redes de área metropolitana. Las redes malladas inalámbricas son una buena opción para este tipo de redes ya que la velocidad de transmisión es superior a otras redes, además no necesita cableado y por tanto se ahorran costes y por último son capaces de alcanzar un área de cobertura extensa.

En los sistemas de salud y medicina son imprescindibles las comunicaciones fiables y rápidas en el entorno del centro para poder compartir archivos y documentos referentes a pruebas médicas, historiales, informes,... Las redes malladas inalámbricas pueden ofrecer mejor servicio para este tipo de aplicaciones que las redes de cable, ya que estas últimas tienen menos flexibilidad para permitir el acceso a todos los posibles dispositivos médicos.

En un edificio donde los dispositivos eléctricos son automatizados para un mejor control y monitorización, si en vez de utilizar una red cableada utilizamos una red mallada ahorramos costes al utilizar routers mallados.

En la actualidad los sistemas de vigilancia se han convertido en una necesidad, y las redes malladas inalámbricas ofrecen la posibilidad de instalarse en cualquier lugar y momento.

Ante una situación de emergencia se puede implementar una red mallada inalámbrica incluyendo un router en el lugar requerido, y de esta manera se establece una red de comunicación rápida.

A continuación explicaremos todas las aplicaciones y el motivo que convierte a las redes malladas inalámbricas en una buena opción para cada una de ellas.



## 2.2. RED DOMÉSTICA DE BANDA ANCHA

Hoy en día podemos encontrarnos en un domicilio con multitud de dispositivos inalámbricos que pueden compartir archivos entre ellos y acceder a Internet. Los dispositivos a los que nos referimos son teléfonos, impresoras, televisores, equipos de música, PDAs y videocámaras. [1]

Para este tipo de redes domésticas las redes malladas inalámbricas son idóneas porque pueden cubrir todas las necesidades del servicio. El principal problema de las redes domésticas son las zonas donde no hay cobertura. Se puede invertir en un estudio de localización de los puntos de acceso pero no es práctico para este tipo de redes ya que resulta ser muy costoso. O también podemos incluir puntos de acceso en las zonas donde no hay cobertura, pero también resultaría costoso. La manera de solucionar estos inconvenientes es utilizar routers mallados como puntos de acceso.

Los routers mallados eliminan las zonas sin cobertura moviendo de posición el router mallado y adecuándolo para que cubra la mayor zona posible. O también ajustando el nivel de potencia de los routers mallados para que su área de cobertura sea mucho mayor. Además los routers mallados nos proporcionan una comunicación entre los nodos más flexible y robusta ante fallos en los enlaces.

Las redes domésticas podrían ser cubiertas por redes ad hoc móviles o redes de sensores, pero eso sobrecargaría cada dispositivo del hogar con la función de encaminamiento.

Otra opción sería las redes Wi-Fi, pero al no ser multisalto perdemos esa flexibilidad de mejora de cobertura.

Concluimos por tanto que la mejor opción para las redes domésticas es utilizar una red mallada inalámbrica de infraestructuras ya que es más económica y cubre las necesidades requeridas como compartir recursos y acceder a Internet.

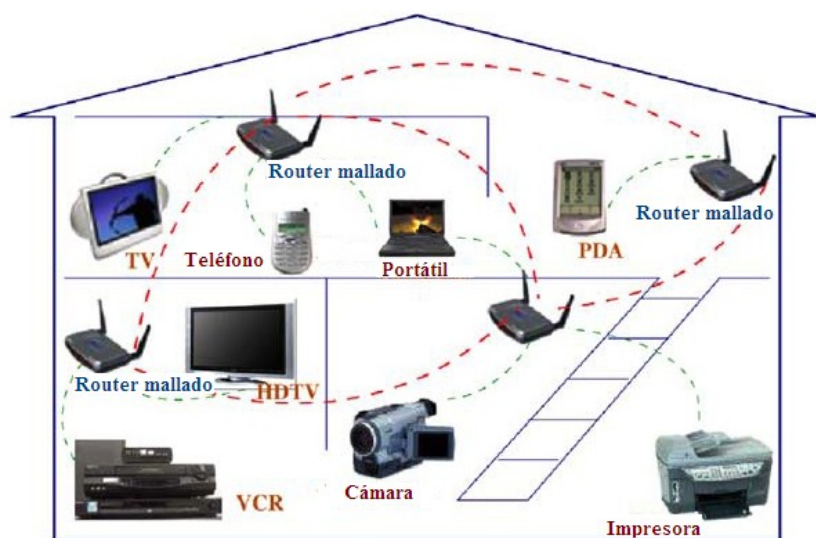


Figura 2.1: Red doméstica de banda ancha [1]

### 2.3. RED MALLADA DE UNA COMUNIDAD DE VECINOS

La arquitectura en las comunidades de vecinos se basa en que la red de acceso es mediante cable o DSL (línea de abonado digital que permite una conexión a una red con más velocidad a través de las líneas telefónicas) conectado a Internet. [1] Este tipo de acceso a la red presenta los siguientes inconvenientes:

- La información de la comunidad viaja por Internet aunque los destinatarios sean sólo los vecinos. Esto reduce la utilización de los recursos de la red.
- Un alto porcentaje de zonas entre las casas no está cubierto por los servicios inalámbricos.
- Resulta costoso ya que el ancho de banda de la gateway no puede ser compartido entre varias casas o vecindarios ya que los servicios inalámbricos son individuales. Como consecuencia se incrementa el coste del servicio de la red.
- Sólo una ruta por casa está disponible para acceder a Internet o conectarse con algún vecino.

La red mallada inalámbrica mitiga las anteriores desventajas gracias a la conectividad flexible. Además la red mallada inalámbrica permite muchas aplicaciones como el almacenamiento de ficheros, acceso a ficheros y video streaming.

En el siguiente dibujo podemos observar un ejemplo de red mallada inalámbrica para una comunidad de vecinos:

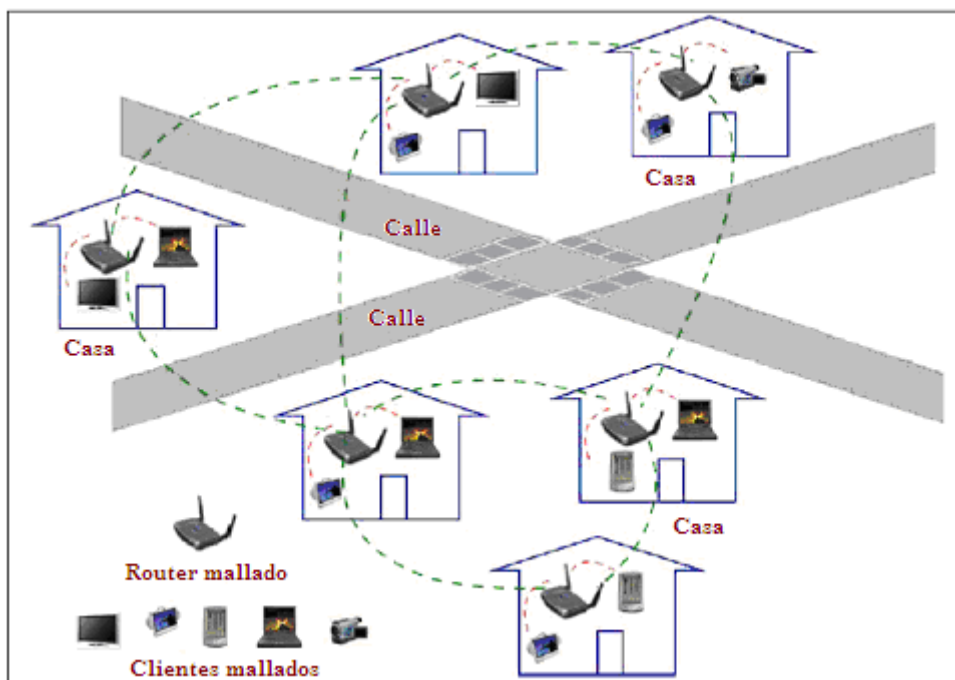


Figura 2.2: Red mallada de una comunidad de vecinos [1]

## 2.4. REDES DE EMPRESAS

Las empresas necesitan de una red inalámbrica que cubra una oficina en el caso de una red pequeña, un edificio de oficinas en el caso de una empresa mediana o un conjunto de edificios de oficinas si el tamaño de la empresa es grande.

Las redes que se utilizan en las empresas utilizan cableado Ethernet para unir a los routers. Como podemos deducir el coste aumenta según el tamaño de la empresa. Además tenemos el mismo problema que en las redes anteriores, hay zonas que no son cubiertas y tendríamos que añadir más routers. Añadiendo más routers mejoramos la capacidad local, pero aumentamos el coste y además no mejoramos la robustez ante fallos en los enlaces y la congestión de la red.

Si utilizamos una red mallada inalámbrica y sustituimos los routers por routers mallados se eliminarían las conexiones de cable Ethernet. Al utilizar routers mallados se aumenta la robustez y la posibilidad de compartir los recursos de la empresa.

Las redes malladas inalámbricas pueden crecer fácilmente hasta el tamaño que la empresa quiera expandir.

Las redes malladas inalámbricas son más complicadas en las redes de empresa que en el caso de redes de comunidad porque el número de nodos es mayor y la topología de la red es más complicada.

Este tipo de redes puede ser utilizado en otros espacios públicos y comerciales como es un aeropuerto, hotel, centros comerciales, pabellón deportivo,...

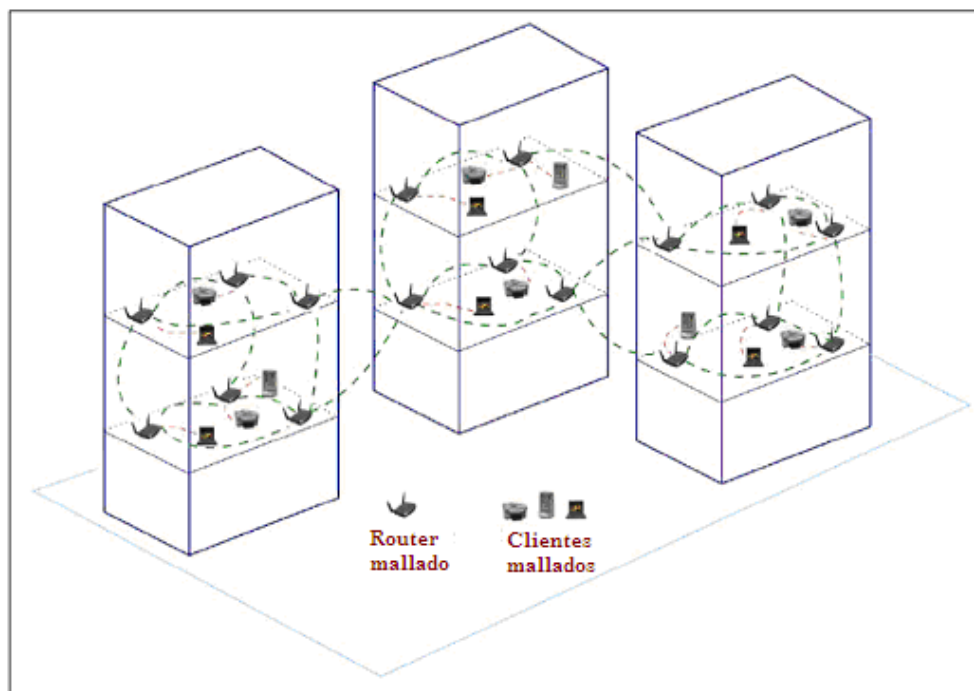


Figura 2.3: Red mallada de empresas [1]

## 2.5. REDES DE ÁREA METROPOLITANA (MAN)

Una red de área metropolitana se define como una red de alta velocidad que cubre un área geográfica extensa. [9] Las redes malladas inalámbricas presentan las ventajas necesarias para este tipo de red.

La velocidad de transmisión es un punto a tener en cuenta, y las redes malladas inalámbricas ofrecen una velocidad superior a otras redes. Además las redes inalámbricas son mucho más económicas que las cableadas y al tratarse de un área geográfica extensa el ahorro es muy grande.

El área de cobertura que abarca la red mallada inalámbrica es mucho mayor al que necesita una red de una casa, edificio, empresa o comunidad. Y la inserción de nuevos nodos para aumentar el área de cobertura se realiza de manera rápida y cómoda.

Por todas estas ventajas que presentan las redes malladas inalámbricas es una opción deseable para abarcar las redes de área metropolitana.

A continuación mostramos un ejemplo del área de cobertura de una red de área metropolitana mallada inalámbrica.

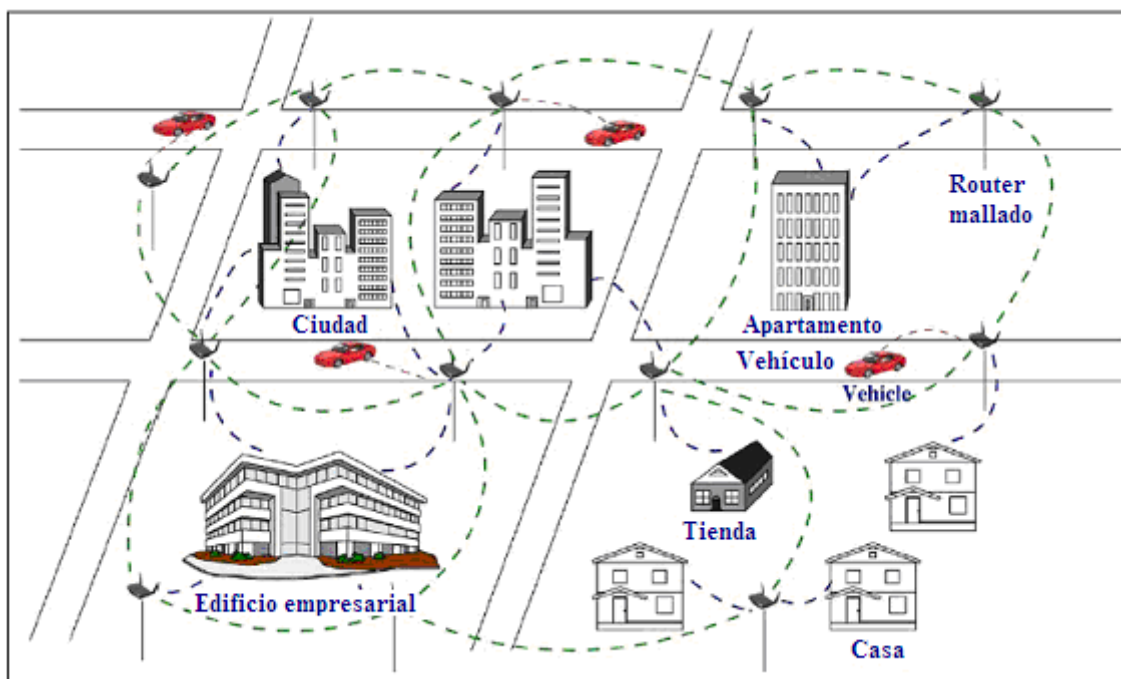


Figura 2.4: Redes de área metropolitana [1]

## 2.6. SISTEMAS DE TRANSPORTES PÚBLICOS

Los sistemas de transportes públicos limitan la red a las estaciones y paradas de transportes públicos. Pero si utilizamos una red mallada inalámbrica estos límites se eliminan y pueden extenderse a vehículos en movimiento como autobuses, ferrys y trenes. De modo que se puede soportar servicios de información al pasajero, monitorización del vehículo mediante video de seguridad y comunicaciones con el conductor. [1]

Para soportar los sistemas de transportes públicos con redes malladas inalámbricas es necesario introducir las redes malladas móviles dentro del vehículo. En el siguiente dibujo vemos un ejemplo:

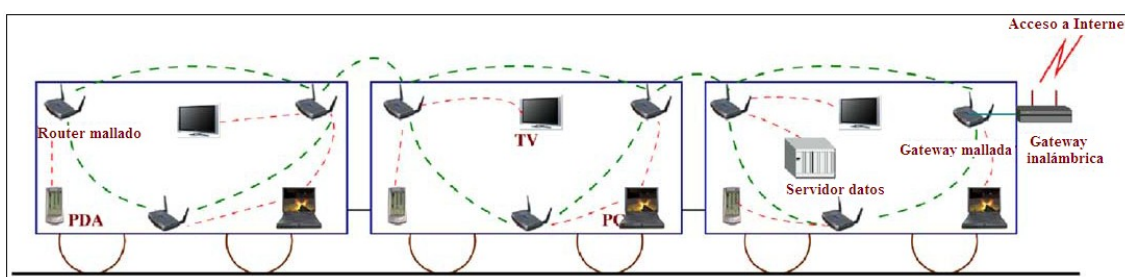


Figura 2.5: Sistemas de transportes públicos [1]

## 2.7. SISTEMAS EN SALUD Y MEDICINA

En el área de la salud y la medicina es importante el uso de redes para compartir archivos y documentos relacionados con pruebas médicas, medicamentos y demás informes necesarios en este área. Por ejemplo en hospitales y centros médicos los datos de monitorización y diagnósticos deben procesarse y transmitirse de una habitación a otra por diversos fines. Esta transmisión de datos debe producirse de manera rápida y asegurando la calidad de las imágenes. Los datos transmitidos en estos centros utilizan banda ancha (broadband), ya que las imágenes médicas tienen una alta resolución y producen un alto volumen de datos constante.

Las redes de cable tradicionales sólo pueden proporcionar acceso a ciertos dispositivos médicos y por este motivo no son muy deseables. Las redes basadas en Wi-Fi se pueden utilizar en más dispositivos y cubren mayor área, pero son más caras y complicadas.

Por este motivo las redes malladas inalámbricas serían deseables en los sistemas de salud y medicina ya que no existen los problemas de las otras redes. [1]

## 2.8. AUTOMATIZACIÓN DE EDIFICIOS

En los edificios nos encontramos con varios dispositivos eléctricos que se automatizan para su control y monitorización. Estos dispositivos son la electricidad, la luz, el ascensor, el aire acondicionado.

En la actualidad esta tarea se realiza a través de redes cableadas. Resulta costoso debido a la complejidad del despliegue y el mantenimiento de las redes cableadas.

Recientemente las redes basadas en Wi-Fi son adoptadas para reducir el coste de las redes cableadas. Pero el coste se reduce aún más si utilizamos redes malladas inalámbricas gracias a la capacidad multisalto que incrementa significativamente la flexibilidad del despliegue.

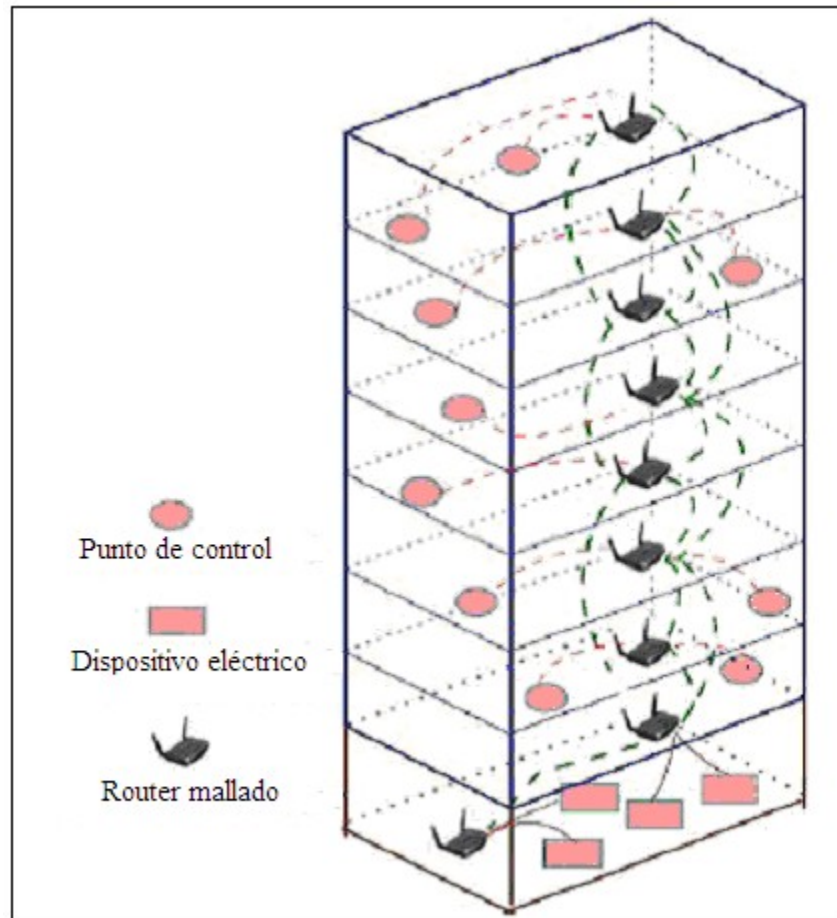


Figura 2.6: Automatización de edificios [1]

## 2.9. SISTEMAS DE VIGILANCIA

Ya que existe un alto grado de preocupación por la seguridad de los establecimientos, se ha convertido en una necesidad el uso de sistemas de vigilancia en centros comerciales y pequeñas tiendas.

Para implementar estos servicios de vigilancia, las redes malladas inalámbricas son más viables que las redes de cable ya que debemos interconectar todos los dispositivos. Y además estos sistemas de seguridad deben instalarse en cualquier lugar que se necesite.

Esta aplicación demanda más capacidad de la red que otras aplicaciones, porque las imágenes fijas y video son el mayor tráfico que circula por la red. [1]

## **2.10. OTRAS APLICACIONES**

Las redes malladas inalámbricas también pueden ser utilizadas en comunicaciones espontáneas requeridas en una emergencia. Con sólo incluir un router mallado inalámbrico en un lugar deseado, la red mallada inalámbrica puede establecer una comunicación rápida.

Para las personas que disponen de dispositivos con conexión inalámbrica, como portátiles y PDAs, pueden comunicarse en cualquier momento y lugar. Y las redes malladas inalámbricas son capaces de satisfacer esta demanda. [1]

## **CAPITULO 3: PROTOCOLOS DE ENCAMINAMIENTO**



### 3.1. INTRODUCCIÓN

La principal función de los protocolos de encaminamiento es seleccionar el camino entre el nodo fuente y destino de una manera rápida y fiable.

Las redes malladas inalámbricas pueden utilizar los protocolos de encaminamiento de otras redes ya existentes, pero modificándolos para que funcione correctamente con las especificaciones de las redes malladas. En esta opción debemos tener en cuenta las características de las redes malladas inalámbricas para centrar los cambios en esta línea.

Si elegimos esta opción, el protocolo de encaminamiento modificado para las redes malladas inalámbricas debe asegurar las principales características que son el número de saltos, el rendimiento, la tolerancia a fallos, el equilibrado de carga, la escalabilidad y el soporte adaptativo. Estudiaremos cada una de las características con mayor detalle en el siguiente punto.

Otra opción es diseñar un nuevo protocolo de encaminamiento para las redes malladas inalámbricas. Esta solución es más costosa ya que cuando se desarrolla un nuevo protocolo hay que probarlo, modificarlo y solucionar los fallos. Por tanto el tiempo de realización es mayor que si nos centramos en un protocolo ya experimentado.

Para diseñar un nuevo protocolo de encaminamiento nos debemos centrar en los factores críticos que influyen en el diseño. Los factores críticos que influyen en las redes malladas inalámbricas son:

- Escalabilidad.
- Conectividad de la malla.
- Banda ancha y QoS.
- Compatibilidad e interoperabilidad.
- Seguridad.
- Facilidad de uso.
- Capacidad.

A continuación estudiaremos la clasificación de los protocolos de encaminamiento que se pueden utilizar para las redes malladas inalámbricas. Nos podemos encontrar con tres grandes grupos, un grupo sería clasificar los protocolos de encaminamiento en base al alcance de las transmisiones, otro grupo sería clasificar los protocolos de encaminamiento en base al modo que descubren las rutas y por último los protocolos de encaminamiento que se basan en el algoritmo que implementan.

Dentro de los protocolos de encaminamiento que se basan en el alcance tenemos dos tipos, unicast y multicast.

En el grupo de protocolos de encaminamiento que se basan en el modo que descubren las rutas, tenemos otros dos grandes grupos que son los que se basan en la topología de la red o los que se basan en la posición de los nodos. Si nos fijamos en los protocolos de encaminamiento que se basan en la topología nos encontramos con los esquemas reactivos y proactivos.

Ejemplos de protocolos de encaminamiento reactivos son AODV, DSR y TORA. Y ejemplos de protocolos de encaminamiento proactivos son B.A.T.M.A.N., OLSR, OSPF, DSDV, PWRP y MMRP.

El protocolo B.A.T.M.A.N. es el que hemos utilizado en el laboratorio para realizar medidas y comprobar su funcionamiento. En el capítulo siguiente se explica con más detalle todas las pruebas y resultados obtenidos.

Y por último tenemos la clasificación de los protocolos de encaminamiento en base al algoritmo que implementan, que puede ser estado del enlace o vector de distancias, según la clasificación de protocolos de encaminamiento de redes tradicionales.

### 3.2. MODIFICACIÓN DE LOS PROTOCOLOS DE ENCAMINAMIENTO DE LAS REDES MÓVILES AD HOC

Para las redes malladas inalámbricas podemos utilizar los protocolos de encaminamiento de otras redes que ya han sido perfeccionados y se encuentran actualmente en funcionamiento. Esta opción es muy aceptada por los clientes ya que prefieren utilizar el mismo protocolo de encaminamiento que ya conocen pero con las modificaciones pertinentes.

En esta línea podemos estudiar los protocolos de encaminamiento de las redes móviles ad hoc. Una red ad hoc es una red sin infraestructura (sin cableado) donde los nodos son móviles y la topología de la red es muy dinámica.

Las redes ad hoc se pueden ver como un caso extremo de redes malladas, a la que todos los nodos son móviles (redes malladas sin infraestructura).

Los protocolos de encaminamiento de las redes ad hoc pueden ser modificados para utilizarlos en redes malladas inalámbricas. Estos cambios se centran en mejorar el rendimiento, la escalabilidad, los requisitos de eficiencia energética y la movilidad, atendiendo a las particularidades de la infraestructura mallada.

Vamos a explicar con más detalle cada una de las características de los protocolos de encaminamiento de las redes ad hoc que son interesantes para redes malladas inalámbricas, y los aspectos a considerar en su aplicación.

- **Número de saltos:** Las redes ad hoc suelen utilizar el número de saltos como una medida para el encaminamiento. Esta medida es adecuada para redes ad hoc ya que las nuevas rutas deben encontrarse de manera rápida, pero no permite distinguir rutas de mejor calidad en un entorno de topologías más estables.

- **Rendimiento:** El rendimiento es una medida que utilizamos para saber si la ruta de encaminamiento es óptima. Supongamos que el camino más corto entre dos nodos es el que presenta una baja calidad, si sólo utilizamos el número de saltos para decidir la ruta, entonces el rendimiento entre estos dos nodos será muy bajo. Para resolver este problema, relacionaremos el número de saltos con la calidad. De modo que si se

produce congestión, entonces el mínimo número de saltos ya no será una correcta medida de rendimiento.

- **Tolerancia a fallos:** Uno de los objetivos para desarrollar en una red mallada inalámbrica es garantizar robustez ante fallos en los enlaces. Si un enlace cae, el protocolo de encaminamiento debe ser capaz de seleccionar rápidamente otra ruta para evitar la interrupción del servicio.
- **Equilibrado de carga:** Uno de los objetivos es compartir los recursos de la red entre muchos usuarios. Cuando una parte de la red experimenta congestión, los nuevos flujos de tráfico no deben ser enviados a través de esas rutas.
- **Escalabilidad:** Es una medida que nos indica la destreza de la red para operar ante cambios topológicos manteniendo la calidad de los servicios. Es una medida importante en topologías dinámicas y en redes que puedan aumentar su tamaño. Para la red mallada es una característica fundamental ya que la topología de la red puede cambiar y debería seguir ofreciendo las mismas calidades ofrecidas antes del cambio.

Por lo tanto, los protocolos de encaminamiento de las redes ad hoc pueden ser utilizados en las redes malladas inalámbricas con algunas consideraciones que acabamos de explicar. Esto supone una gran ventaja ya que la infraestructura está desplegada y las operadoras prefieren utilizar un protocolo que ya conocen (aunque modificado), que uno totalmente novedoso. Además si trabajamos sobre un protocolo ya creado obtenemos una base sólida de conocimientos y experiencias a partir de la cual trabajar. Es mejor ampliar un protocolo conocido, en vez de desarrollar un nuevo protocolo que debe ser probado, modificado y solucionado, y esto puede llevar mucho tiempo.

### 3.3. DISEÑO DE NUEVOS PROTOCOLOS DE ENCAMINAMIENTO

Como ya hemos explicado en el punto anterior los protocolos de encaminamiento de las redes móviles ad-hoc se pueden utilizar en redes malladas inalámbricas, con ciertas adaptaciones para atender a las particularidades del entorno mallado con infraestructura, aunque en ciertos casos se podría diseñar un protocolo de encaminamiento desde el cero.

Antes de diseñar, desarrollar u operar con una red, hay que considerar los factores críticos que influyen en el diseño de los protocolos de encaminamiento. [1] Para una red mallada inalámbrica, son los siguientes:

- Escalabilidad:

Para la creación de redes multi-salto, hay que saber que los protocolos de encaminamiento sufren problemas de escalabilidad, es decir, cuando el tamaño de la red aumenta, el rendimiento de la red se degrada considerablemente.

Existen otras situaciones que afectan a la escalabilidad, como que los protocolos de encaminamiento no sean capaces de encontrar una ruta fiable o que los protocolos de transporte pierdan las conexiones.

El motivo de la baja escalabilidad es debido a que la fiabilidad extremo a extremo cae bruscamente frente a la extensión de la red.

- Conectividad de la malla:

Muchas de las ventajas de las redes malladas inalámbricas provienen de la conexión mallada que interconecta a todos los nodos que forman la red. Los nodos pueden estar interconectados todos con todos o con la gran mayoría. Este requisito es indispensable y además debemos asegurar que los nodos sean capaces de auto-organizarse.

- Banda ancha y QoS:

La diferencia con otras redes es que la mayoría de las aplicaciones de las redes malladas inalámbricas son servicios de banda ancha con diferentes requisitos de QoS. Por lo que los protocolos de encaminamiento de las redes malladas inalámbricas deben considerar el retardo extremo a extremo, el rendimiento y la pérdida de paquetes.

- Compatibilidad e interoperabilidad:

Se trata de una característica deseada para las redes malladas inalámbricas que debe soportar el acceso a clientes convencionales de otras redes y a los propios clientes de la red mallada. Por lo tanto, las redes malladas inalámbricas necesitan ser compatibles con los nodos convencionales, de lo contrario, la motivación de desplegar esta red será mínima. La integración de la red mallada inalámbrica con otras redes inalámbricas requiere de routers mallados que tengan capacidad de interoperar con otras redes inalámbricas.

- Seguridad:

Una parte muy importante para que las redes malladas inalámbricas sean acogidas con convicción es que ofrezcan seguridad al cliente. Sin una solución de seguridad sólida, este tipo de redes no tendrán éxito debido a la falta de incentivos por parte de los clientes a suscribirse a servicios poco fiables.

Aunque se han propuesto muchos esquemas de seguridad para LANs inalámbricas, todavía no hay ninguno listo para redes malladas inalámbricas. Por ejemplo, no hay una autoridad centralizada de confianza para distribuir una clave pública en una red mallada inalámbrica.

Existen esquemas de seguridad propuestos para redes ad hoc que pueden ser adoptados para redes malladas inalámbricas, pero existen varios inconvenientes:

- La mayoría de las soluciones de seguridad para redes ad hoc todavía no están lo suficientemente evolucionadas como para ser aplicadas.
- La arquitectura de la red mallada inalámbrica es diferente a la red convencional ad hoc, lo que provoca diferencias en los mecanismos de seguridad.

Como consecuencia, los nuevos esquemas de seguridad deben desarrollar:

- Algoritmos de cifrado que distribuyan de manera segura la clave.
- MAC y protocolos de encaminamiento seguros.
- Detección de intrusiones, seguridad y vigilancia.

- Facilidad de uso:

Los protocolos deben ser diseñados para permitir que la red sea lo más autónoma posible, en el sentido de gestión de capacidades, auto-organización, control dinámico de la topología, robustez en el tiempo de fallo y rapidez en la autenticación de suscripciones de usuarios a la red. Además, es necesario desarrollar herramientas de gestión de red para mantener la operación, supervisión de la ejecución y configuración de los parámetros de las redes malladas inalámbricas. Estas herramientas junto a la autonomía de los mecanismos en los protocolos permitirá el despliegue rápido de las redes malladas inalámbricas.

- Capacidad:

Una buena comprensión entre la capacidad de la red y los factores explicados anteriormente, nos guían para la elaboración de los protocolos de encaminamiento, diseño de la arquitectura y operación de la red.

La capacidad en las redes malladas inalámbricas está afectada por varios factores como:

- La arquitectura de la red.
- La topología de la red.
- El tipo de tráfico.
- La densidad de nodos de la red.
- El número de canales utilizados en cada nodo.
- El nivel de potencia de transmisión.
- La movilidad del nodo.

Muchas investigaciones se han centrado en estudiar la capacidad de las redes ad hoc, para así investigar la capacidad de la redes malladas inalámbricas. [9]

De los resultados analíticos se deduce que el rendimiento de la capacidad se reduce significativamente cuando la densidad de los nodos aumenta.

### 3.4. CRITERIOS PARA LA CLASIFICACIÓN DE PROTOCOLOS

Para poder decidir el protocolo de encaminamiento que más se adecue a las necesidades de la red mallada inalámbrica que estamos creando, nos debemos fijar en su funcionamiento. Y para ello podemos clasificar los protocolos de encaminamiento en los siguientes criterios:

- Según el alcance:

Se distingue dos tipos de protocolos si nos basamos en el alcance: **unicast** y **multicast**.

Los protocolos unicast son los que transmiten la información de un único origen a un único receptor.

En los protocolos multicast el emisor transmite la información a un grupo de nodos que están dentro del grupo multicast. Dentro del protocolo multicast existen varios casos como el broadcast, geocast y anycast. El protocolo broadcast envía la información a todos los nodos de la red. El geocast transmite a los nodos que se sitúan en la misma zona geográfica. Y el anycast envía a un nodo dentro de un grupo.

- Según el modo que se descubren las rutas:

Dentro de los protocolos de encaminamiento que se basan en el modo que descubren las rutas nos podemos encontrar con dos tipos:

- Protocolos basados en la topología.
- Protocolos basados en la posición.

Los protocolos de encaminamiento basados en la topología se basan en la información topológica de la red para seleccionar la trayectoria. Para descubrir las rutas hacia los destinos de la red se usan los esquemas **proactivos y reactivos**.

Los protocolos proactivos persiguen tener un conocimiento exhaustivo del estado de la red. De modo que cuando se necesita una ruta, ésta ya es conocida y está lista para usarse de manera inmediata. En escenarios cambiantes no es muy aconsejable, porque se precisa que las tablas de encaminamiento estén actualizadas mediante el envío continuo de mensajes. Esto provoca una sobrecarga de mensajes de control en la red. Ejemplos de este tipo de protocolos son B.A.T.M.A.N., OLSR, OSPF, DSDV, PWRP y MMRP.

Los protocolos reactivos sólo obtienen información de encaminamiento cuando es necesario. En consecuencia la sobrecarga de la red es menor que en los protocolos proactivos. Pero en cambio el tiempo en establecer la comunicación aumenta. Ejemplos de este tipo de protocolos son AODV, DSR y TORA.

También existen protocolos que combinan los anteriores y son los que denominamos protocolos de encaminamiento híbridos. Utilizan las ventajas de cada uno de los protocolos, por un lado utiliza el encaminamiento proactivo cuando los nodos están cerca y el encaminamiento reactivo cuando los nodos están lejos. También utilizamos el encaminamiento reactivo cuando los caminos son utilizados en pocas ocasiones.

Los protocolos de encaminamiento basados en la posición se basan en las informaciones geográficas para seleccionar la trayectoria. Los paquetes son enviados basándose en la posición geográfica de los nodos a atravesar, los nodos vecinos y el nodo destino. Para ello cada nodo debe conocer su posición geográfica. La posición del destino tiene que ser proporcionada por un servidor de localización. Un ejemplo de este tipo de protocolos de encaminamiento es GPSR.

- Según el tipo de algoritmo que implementan:

Si nos fijamos en el tipo de información que se utiliza podemos hablar de protocolos que se fijan en el **estado de los enlaces** o en el **vector de distancia**.

En los protocolos de vector de distancia cada nodo presenta una tabla de encaminamiento con el enlace y el coste asociado a cada nodo. Cada cierto tiempo el nodo transmite su tabla de encaminamiento a sus vecinos y éstos recalculan su tabla de encaminamiento si existe nueva información.

En los protocolos de estado de enlace todos los nodos tienen una tabla con el mapa de red completo. En esta tabla se define el enlace y la distancia para llegar de un nodo a otro. Y cada nodo envía cada cierto tiempo la información de cómo llegar a sus vecinos.

### **3.5. EJEMPLOS DE PROTOCOLOS DE ENCAMINAMIENTO PARA LAS REDES MALLADAS INALÁMBRICAS**

Vamos a explicar en detalle algunos protocolos de encaminamiento:

- Proactivos: B.A.T.M.A.N., OLSR, OSPF, DSDV, PWRP y MMRP.
- Reactivos: AODV, DSR y TORA
- Basado en posición: GPSR.

Veremos el funcionamiento de cada uno de ellos y las características que lo hacen especial para las redes malladas inalámbricas.

#### **3.5.1. PROTOCOLOS DE ENCAMINAMIENTO PROACTIVOS: B.A.T.M.A.N.**

B.A.T.M.A.N. es un protocolo de encaminamiento dinámico y proactivo para redes malladas ad-hoc que utiliza las tablas de encaminamiento para las decisiones de encaminamiento. Este protocolo no calcula rutas completas entre un nodo origen y destino sino que selecciona un nodo de salto para utilizarlo como gateway hacia el destino.

La función de B.A.T.M.A.N. es encontrar otros nodos B.A.T.M.A.N. y definir el mejor vecino para llegar a ellos. Además hace un seguimiento de los nuevos nodos e informa a sus vecinos de su existencia.

Es decir, cuando un nodo se incorpora a la red envía un paquete broadcast para avisar de su existencia. Este mensaje se va distribuyendo por toda la red. Cuando un nodo recibe este mensaje anota la dirección por donde le ha llegado la información más rápido, y así si tiene que enviarle información utilizará ese nodo (gateway) para transmitir. El protocolo mantiene la información sobre la existencia de los nodos mientras sean accesibles. Por lo tanto el protocolo B.A.T.M.A.N. busca un nodo vecino hacia el destino para ser utilizado como gateway para conseguir la comunicación, es decir, se encarga de elegir el mejor salto para cada destino. Esto provoca que la implementación sea rápida y eficiente, ya que no es necesario calcular la ruta completa.

Cuando es necesario decidir por qué ruta transmitir se realizan análisis estadísticos de la pérdida de paquetes y de la velocidad de propagación de esa ruta. No se basan en el estado de los nodos ni en la información topológica de otros nodos.

Los paquetes de datos del protocolo B.A.T.M.A.N. contienen muy poca información, lo que provoca que sean muy pequeños y que su transmisión sea rápida.

Cada nodo percibe y mantiene la información sobre el mejor salto hacia el resto de nodos. Por este motivo es innecesario el conocimiento global de la red.

Resumiendo, cada nodo transmite mensajes broadcast para indicar a los nodos vecinos de su existencia. Y los vecinos lo retransmiten a sus vecinos, y así sucesivamente. De este modo el mensaje llega a todos los nodos de la red. Estos mensajes son pequeños, el tamaño típico es de 52 bytes. Los campos que contienen estos mensajes son la dirección de origen, la dirección del nodo que retransmite, el TTL y el número de secuencia.

Es posible que se reciban varias veces el mismo mensaje, ya que hay rutas donde se propagan con retardo o con pérdidas debido a la baja calidad del enlace, y habrá otras donde se propagan de manera rápida y fiable. Para solucionar este problema se etiqueta cada mensaje con un número de secuencia. De este modo se hace un estudio de que nodo vecino es el que le ha transmitido la información primero y lo selecciona como el vecino de salto para enviar datos al origen del mensaje y lo configura en su tabla de encaminamiento.

Las principales características de este protocolo son:

- **Soporte en múltiples interfaces:** se puede utilizar en más de una tarjeta inalámbrica o ethernet.
- **Soporte de interfaces alias:** lo que permite ejecutar otros protocolos en paralelo.
- **Interfaz IPC** (Inter-Process Communication): para conectarse al demonio batmand. Algunas de las acciones que permite son:
  - Información debugging
  - Consulta de nodos vecinos
  - Consulta de nodos gateway
  - Configuración del gateway en tiempo de ejecución
  - Mensajes HNA
  - Modificación de interfaces
- **Policy Routing:** A partir de la versión B.A.T.M.A.N. 0.3 se soporta el policy routing y se puede utilizar las funciones especiales de encaminamiento proporcionadas por el kernel de Linux. Con el comando “route” se puede consultar las 4 tablas de encaminamiento. B.A.T.M.A.N.:
  - Tabla “redes”: Contiene las entradas de las redes anunciadas (HNA).
  - Tabla “hosts”: Contiene todos los nodos alcanzables.



- Tabla “unreachable”: Contiene los nodos que no son alcanzables en la red.
- Tabla “tunnel”: Contiene la ruta por defecto que se utiliza si existe un gateway disponible.
- **Selección de gateway:** El protocolo B.A.T.M.A.N. permite anunciar una conexión a Internet, e informa del ancho de banda disponible, la velocidad de bajada y de subida. Podemos buscar anuncios de gateway y decidir a cuál conectarnos basándonos en los siguientes modos de comportamiento:
  - Fast Internet Connection: Considera la calidad del enlace y la clase de gateway. Mantiene la conexión hasta que ya no se puede.
  - Stable Internet Connection: Elige la conexión más estable hacia la gateway y la mantiene hasta que ya no se puede.
  - Fast-switching: Elige la conexión más estable hacia la gateway pero la cambia cuando encuentra otra mejor.
  - Late-switching: Elige la conexión más estable hacia la gateway pero la cambia cuando encuentra otra conexión con X veces más calidad. (X parámetro configurable).

El cliente de Internet B.A.T.M.A.N. puede detectar y evitar gateways que posean enlaces que están fallando. Además durante el tiempo de ejecución en Internet se puede desactivar la gateway, cambiar a otra gateway o la clase de encaminamiento, sin necesidad de reiniciar el demonio.

- **Servidor de visualización:** Ya que el protocolo B.A.T.M.A.N. no contiene la topología de la red, cada nodo envía su vista local de sus vecinos al servidor de visualización y éste con todos los envíos de cada nodo hace una recopilación y los incluye en el formato que utilizan las herramientas de visualización de red.

Además el protocolo B.A.T.M.A.N. puede ser utilizado en las redes cableadas, como Ethernet.

La idea de crear este protocolo es para que fuese fácil, rápido y ligero. Por lo que se dividió el desarrollo en varias fases para ejecutar funciones complejas mediante un proceso de desarrollo por fases [17]:

- Versión 1: La primera fase se centra en implementar el protocolo B.A.T.M.A.N. y comprobar que funciona correctamente. Se prueba el envío de mensajes broadcast y recepción de los mismos. Y que las tablas de encaminamiento se rellenen según las especificaciones del protocolo, es decir, que incluya el nodo destino y el vecino de salto.
- Versión 2: En la segunda fase se centra en mejorar los datos de la tabla de encaminamiento, asegurando que las entradas son válidas. Esto se consigue con mecanismos que acepten la información de los nodos que tengan enlaces bidireccionales.

- Version 3: En la última fase se ofrece la posibilidad de que el protocolo B.A.T.M.A.N. sea soportado en múltiples dispositivos de la red. Este hecho supone un gran avance para poder distribuir el protocolo ya que los costes serían bajos ya que disponemos de dispositivos como ordenadores y portátiles distribuidos entre los clientes. Y además los clientes lo aceptarían con más facilidad ya que utilizan dispositivos con los que están familiarizados.

En esta fase también se incluye la posibilidad de acceder a Internet cuando un nodo de la red mallada tiene acceso. El proceso es el siguiente, el nodo que tiene acceso a Internet se anuncia como gateway e informa de la velocidad y el ancho de banda que ofrece. Los nodos que reciben el aviso desde ese momento pueden acceder a Internet a través de esa gateway. Si se recibe más de una gateway se decide según los parámetros de velocidad y estabilidad.

### **3.5.2. PROTOCOLOS DE ENCAMINAMIENTO PROACTIVOS: OLSR**

OLSR [4] es un protocolo proactivo que se basa en el estado de los enlaces. Es una optimización del algoritmo de estado de enlaces pero reduciendo el tamaño de las tablas de encaminamiento y el número de retransmisiones de mensajes broadcast.

Los nodos se intercambian mensajes de Hello para saber que están en funcionamiento. Se utiliza la técnica MPR (Multipoint Relaying) que consiste en elegir un conjunto de nodos vecinos que cubran el acceso de nodos distantes a 2 saltos o más. OLSR se adapta bien en redes con un gran número de nodos y de alta movilidad.

OLSR utiliza el encaminamiento salto a salto, es decir, utiliza su información local para encaminar los paquetes.

Este protocolo no hace uso de una entidad central, sino que actúa de manera distribuida. No hace falta asegurar una transmisión periódica de mensajes de control, se puede permitir alguna pérdida. Tampoco se necesita una recepción de mensajes secuencial, se usan números de secuencia que se incrementa en cada mensaje para que el nodo receptor sea consciente de que información es más reciente.

A continuación vamos a explicar las palabras claves para entender el funcionamiento de este protocolo:

- **Nodo**: Router que implementa el protocolo OLSR.
- **Interfaz OLSR**: Interfaz de un equipo que participa en el protocolo OLSR.
- **Dirección principal**: Dirección origen del tráfico de control.
- **Nodo vecino**: Un nodo es vecino de otro nodo, si se pueden escuchar entre ellos. Dos nodos son vecinos si se encuentran en la misma área de cobertura.

- Vecino a 2 saltos: Nodo que no es vecino, pero que está dentro del área de cobertura de un nodo vecino.
- Vecino a 2 saltos estricto: Nodo que es vecino de un vecino.
- MPR (Multipoint relay): Nodo seleccionado por su vecino, nodo X, encargado de retransmitir todos los mensajes broadcast que reciba del nodo X.
- MPR selector (MS): Nodo que ha seleccionado a un vecino como MPR.
- Enlace: pareja de interfaces OLSR sensibles de escucharse uno al otro. Los enlaces pueden ser simétrico (enlace bidireccional) o asimétricos (sólo verificados en un sentido).
- Vecindario simétrico de 1 salto: En un nodo X es el grupo de nodos que tiene un enlace simétrico hacia X.

El protocolo OLSR tiene un núcleo de funcionalidades que siempre es requerido, además de un grupo de funcionalidades auxiliares.

#### **Funcionamiento del núcleo:**

El núcleo especifica el comportamiento de un nodo que tiene interfaces OLSR. Se basa en las siguientes funcionalidades:

- Formato de paquete y retransmisión: El formato del paquete es igual para todos los datos del protocolo OLSR, así es fácil la extensión de este protocolo. Los paquetes se envían como datagramas UDP. El nodo descarta el paquete en las siguientes situaciones:
  - o Si el paquete no contiene mensaje (el tamaño es demasiado pequeño).
  - o Si el valor del TTL es menor o igual a 0.
- Condiciones de retransmisión: Un mensaje si es un duplicado no se retransmite.
- Percepción del enlace: Para saber el estado de un enlace se envían mensajes de HELLO. Cada nodo tiene asociado a cada vecino el estado del enlace.
- Detección de vecino: Cuando un nodo detecte la aparición de un nuevo vecino se debe incluir una nueva entrada a la tabla de encaminamiento e incluir el estado del enlace. Además si se detecta una variación en el estado de un enlace, se debe comprobar en la tabla de encaminamiento que el cambio ha sido reflejado. Si no se recibe información de un enlace durante un tiempo determinado se elimina de la tabla de encaminamiento el enlace y el vecino correspondiente.
- Selección y señalización de MPR: Los MPRs sirven para seleccionar los nodos vecinos que van a realizar el broadcast de los mensajes de control. La señalización se realiza mediante mensajes de HELLO.

- Difusión de mensajes de control de la topología: Estos mensajes son denominados TC (Topology Control) y contienen información sobre los enlaces de los vecinos que puede alcanzar. Este mensaje se envía a los MPR para que el resto de nodos conozcan a los vecinos que puede alcanzar el nodo origen.
- Cálculo de rutas: Cada nodo contiene una tabla de encaminamiento con el estado del enlace y el nodo. El estado del enlace se mantiene gracias al intercambio de mensajes periódicos. La tabla de encaminamiento se actualiza si se detecta algún cambio en el campo de enlace, de vecino, de vecino de dos saltos o en la topología.

### **Funciones auxiliares:**

Existen situaciones donde es necesario utilizar funciones auxiliares, por ejemplo en un nodo con múltiples interfaces donde algunas de ellas se conectan a otras redes. Para conseguir la conectividad entre las interfaces OLSR y el resto de interfaces debemos introducir información externa del encaminamiento de la red. Para conseguir esto las interfaces no OLSR crean un mensaje HNA (Host and Network Association) que contiene la información necesaria para poder crear nuevas rutas.

OLSR no trabaja con la información de la capa de enlace, pero si está disponible se utiliza para mantener la información de los vecinos y de los MPR. Para proveer de redundancia a la red se emite información del nodo con todos los enlaces hacia sus vecinos. Existen tres posibles niveles de redundancia:

- Sin redundancia: La información se emite a los nodos que le han elegido como MPR.
- Redundancia media: Además de emitir a los nodos que le han elegido como MPR emite a los nodos que él ha elegido como MPR.
- Redundancia alta: Se emite información de todos los enlaces hacia los vecinos.

Otra funcionalidad es que el nodo puede seleccionar MPRs redundantes. A pesar que se crea más tráfico y se pierde eficiencia, se tiene una gran ganancia al asegurar la llegada de los paquetes a sus destinos. Esta funcionalidad es útil en redes con mucha movilidad, ya que mantiene una buena cobertura con los MPRs.

### **3.5.3. PROTOCOLOS DE ENCAMINAMIENTO PROACTIVOS: DSDV**

DSDV (Destination Sequenced Distance Routing) es un protocolo unicast proactivo adaptado del tradicional protocolo RIP (Routing Information Protocol). Añade al protocolo RIP el número de secuencia, que es un nuevo atributo que se incluye en la tabla de encaminamiento. Esta información es útil para detectar la información más reciente y para evitar bucles. La tabla de encaminamiento de cada nodo presentan los siguientes datos: posibles destinos, el siguiente salto, el número de secuencia, el coste y el TTL.

La tabla de encaminamiento se actualiza cuando se reciben mensajes periódicos de nodos nuevos o cuando se produce un cambio en la topología. Para mantener una información coherente en las tablas de encaminamiento, cuando se produce un cambio en la topología se envían mensajes con la nueva información de encaminamiento de manera rápida y frecuente.

#### **3.5.4. PROTOCOLOS DE ENCAMINAMIENTO PROACTIVOS: OSPF**

OSPF (Open Short Path First) es un protocolo de encaminamiento basado en el estado de enlace. Se puede utilizar en redes pequeñas y grandes. En redes grandes se utiliza el diseño jerárquico. Varias zonas se conectan a un área de distribución o área cero que se denomina backbone. Definiendo estas áreas se consiguen las siguientes ventajas.

- Reduce el gasto de procesamiento de información.
- Acelera la convergencia.
- Limita la inestabilidad de la red a un solo área.
- Mejora el rendimiento.

Cada nodo OSPF contiene la información de los nodos vecinos con su correspondiente estado de enlace. Y esta información es enviada a todos los vecinos. De modo que un nodo OSPF publica sus estados de enlace y los enlaces recibidos. Así cada nodo del área está informado de la base de datos y vecinos del resto de nodos.

Para reducir el número de mensajes de encaminamiento entre los vecinos de la misma red, se selecciona un router designado (DR) y un router designado de respaldo (BDR) que sirven como intercambiador de mensajes de información de encaminamiento.

Existen cinco tipos de mensajes:

→ Hello: Que sirve para identificar a los vecinos.

→ Database Description Packets: Intercambio de información entre dos routers en el momento que se ha establecido la comunicación, para que otros routers puedan descubrir los datos que les faltan durante la inicialización o sincronización.

→ Link State Request: Esta petición se realiza si durante el intercambio de información entre dos routers uno detecta que le faltan datos o que la información está obsoleta.

→ Link State Reply: Es la respuesta a la petición del estado de los enlaces. También se utiliza para informar periódicamente de los cambios en la topología. El emisor retransmitirá esta respuesta hasta que se reciba un mensaje ACK de confirmación.

→ Link State ACK: Confirma la recepción de un Link State Reply,

Las principales características de OSPF es que ofrece una respuesta rápida y sin bucles ante cambios de topología, soporta múltiples métricas y balancea la carga en múltiples caminos.

### **3.5.5. PROTOCOLOS DE ENCAMINAMIENTO PROACTIVOS: PWRP**

PWRP (Predictive Wireless Routing Protocol) es el único protocolo de encaminamiento desarrollado específicamente para redes inalámbricas. Además de validar el estado del enlace, como todos los demás protocolos, valida la calidad del enlace.

Este protocolo recalcula 4 veces por segundo el estado de los enlaces de toda la red mallada. Y selecciona la ruta en función del estado de la red mallada en cada momento. Este funcionamiento soporta las redes dinámicas y nos proporciona la ruta más óptima en cada momento.

### **3.5.6. PROTOCOLOS DE ENCAMINAMIENTO PROACTIVOS: MMRP**

MMRP (Multiple Metric Routing Protocol) [8] denominado MM, utiliza en el proceso de descubrimiento una inundación de paquetes de búsqueda de ruta. El nodo destino responde al paquete que contenga un coste acumulativo menor.

Existen variantes multi-salto que intentan reducir la inundación de mensajes, ya que provocan congestión en el canal y consumo energético.

- La primera propuesta se denomina D\_MM (Direccionalidad) y utiliza para el proceso de búsqueda de una ruta el conocimiento de la ubicación del nodo destino. De modo que cada salto supone una progresión hacia el nodo destino y por tanto se reduce la inundación de mensajes. Y de todas las rutas disponibles hacia el nodo destino se elija aquella que minimice la función de coste.
- La segunda propuesta se denomina Dx-MM (Direccionalidad con x-permisos) y considera la dirección entre los nodos origen y destino en el proceso de búsqueda de la ruta. El paquete RREQ contiene un campo que indica el número de saltos permitidos en la dirección contraria al destino. El paquete RREQ se propaga a medida que la ruta avanza hacia el nodo destino y siempre que no se sobrepase el número de saltos permitidos por el nodo origen.
- La tercera propuesta se denomina hrD-MM (Direccionalidad híbrida) y busca el compromiso entre el rendimiento del algoritmo MM y la eficiencia de D-MM. Primero se realiza la búsqueda de la ruta utilizando el algoritmo D-MM y si no es capaz de crear una ruta se utiliza el algoritmo MM. Y de esta manera se aumenta las probabilidades de creación de la ruta.

Es un protocolo muy bueno para entender el encaminamiento mallado y se puede implementar fácilmente con laptops ejecutando Linux.

### **3.5.7. PROTOCOLOS DE ENCAMINAMIENTO REACTIVOS: AODV**

AODV [4] es un protocolo diseñado para redes móviles autoconfigurables con demanda del vector de distancia. Permite el encaminamiento dinámico, autoconfigurable y multi-salto entre nodos.

Se trata de un protocolo reactivo y unicast que se construye sobre el protocolo proactivo DSDV (que antes hemos explicado). La mejora que realiza sobre el protocolo DSDV es que minimiza el número de broadcast requeridos para crear rutas. Esto es así porque al tratarse de un protocolo bajo demanda los nodos que no están en el camino elegido no tienen que mantener la ruta ni participar en el intercambio de las tablas de encaminamiento.

Cuando un nodo quiere transmitir y no encuentra una ruta válida en su tabla de encaminamiento comienza con el proceso de descubrimiento (Path Discovery). Entonces se realiza un broadcast de mensajes RREQ a sus vecinos hasta que alcance al destino o a algún nodo intermedio que tenga la ruta hacia el destino creada recientemente. Para identificar si las rutas son recientes utilizamos los números de secuencia.

Cuando el mensaje de RREQ llega al destino o a un nodo con una ruta reciente hacia el destino, responde enviando un mensaje de RREP al vecino del que recibió el primer RREQ. El mensaje de RREP contiene los siguientes campos:

- Dirección origen.
- Dirección destino.
- Número de saltos.
- Número de secuencia del destino.
- Tiempo de expiración para el camino de vuelta.

El mensaje RREP utiliza el camino establecido desde el origen al nodo que envía el RREP y todos los nodos intermedios anotan la ruta como la más reciente hacia el destino. Por este motivo AODV sólo puede emplearse en enlaces bidireccionales.

También se utilizan mensajes de Hello (broadcast periódicos) para informar a un nodo móvil sobre todos los nodos vecinos. Si un enlace se rompe, el nodo que detecta la caída envía un mensaje de Route Error. La manera de detectar un enlace caído puede ser por el continuo envío de mensajes de Hello para confirmar la conectividad. Si en un tiempo establecido no se recibe ningún mensaje de Hello, se asumirá que el enlace se ha caído. Otro mecanismo para detectarlo es la transmisión sin éxito de las tramas.

### **3.5.8. PROTOCOLOS DE ENCAMINAMIENTO REACTIVOS: DSR**

En el protocolo DSR [4] el encaminamiento se organiza desde el origen. Se incluye en la cabecera de los datos un campo de información sobre los nodos exactos que debe atravesar. De modo que no se necesita de mensajes periódicos y se disminuye la sobrecarga de mensajes de control. Además ofrece la posibilidad de obtener con una solicitud de una ruta, múltiples caminos hacia el destino.

Cada nodo dispone de una memoria caché de rutas donde almacena las rutas descubiertas, ya sea por procesos propios del nodo o por escuchas en la red.

En el proceso de descubrimiento de rutas intervienen los siguientes mensajes:

- ROUTE REQUEST: Mensaje de solicitud (RREQ).
- REPLY: Mensaje de respuesta (RREP).
- ERROR: Mensaje de error.

El protocolo presenta dos fases: descubrimiento y mantenimiento de rutas. Cuando un nodo quiere transmitir lo primero que hace es consultar su tabla de encaminamiento para saber si hay una ruta hacia ese destino. Si no tiene la ruta comienza el descubrimiento mediante broadcast de mensajes RREQ. Este paquete contiene:

- La dirección de destino.
- La dirección del nodo origen.
- El número de identificación.

El mensaje RREP se envía cuando el mensaje RREQ ha llegado al destino o algún nodo intermedio que disponga de una ruta hacia el destino.

Se realiza un mantenimiento de rutas para monitorizar el funcionamiento de la ruta e informar al nodo origen sobre cualquier error en el encaminamiento. Los mensajes de error (Router Error Packet) se producen cuando la capa de enlace detecta un problema grave de transmisión.

### **3.5.9. PROTOCOLOS DE ENCAMINAMIENTO REACTIVOS: TORA**

TORA (Temporally Ordered Routing Algorithm) [\[12\]](#) [\[13\]](#) es un protocolo de encaminamiento reactivo que se basa en ofrecer al nodo origen múltiples trayectos para alcanzar al destino.

Este protocolo fue propuesto para redes muy dinámicas. Cuando se detecta un cambio en la topología se generan mensajes de control en un pequeño conjunto de nodos cercanos al cambio.

Para utilizar estas funciones básicas TORA define tres tipos de paquetes de control:

- QUERY (QRY)
- UPDATE (UPD)
- CLEAR (CLR)

En las fases de creación y mantenimiento el paquete QRY se utiliza como petición y el paquete UPD se utiliza como respuesta. Cuando un nodo necesita descubrir una ruta hacia un destino se hace un envío broadcast de mensajes QRY. El destino o un nodo que posea una ruta válida hacia el destino le responde con un mensaje UPD. Y el paquete CLEAR se utiliza como broadcast para eliminar las rutas inválidas cuando un nodo es inaccesible.



### 3.5.10. GPSR (Greedy Perimeter Stateless Routing)

GPSR [4] [10] es uno de los primeros protocolos de encaminamiento basados en posición prácticos para las redes inalámbricas. Este protocolo reacciona rápidamente y es eficiente en redes móviles.

GPSR estudia la relación entre la posición y la conectividad geográfica de la red inalámbrica, utilizando las posiciones de los nodos para tomar decisiones de retransmisión de paquetes. Por tanto para decidir la transmisión el nodo sólo utiliza la información de los vecinos inmediatos. Cuando un paquete llega a una región en la que la transmisión es imposible, el algoritmo de encaminamiento recupera el enrutamiento alrededor del perímetro de la región. Cuando hay cambios frecuentes en la topología, GPSR puede usar la información de la topología local para encontrar una nueva ruta para transmitir de manera rápida.

GPSR utiliza greedy forwarding que consiste en retransmitir el paquete a nodos que están más cercanos al destino que el nodo que retransmite. Si no existe un camino greedy, que permite un salto a un nodo más cercano al destino, se utiliza perimeter mode, en el cual se atraviesan caras sucesivas del plano hasta que alcanza un nodo cercano al destino, y a partir de aquí se vuelve a utilizar greedy forwarding. Es decir, GPSR consta de dos métodos de transmisión de paquetes: **greedy forwarding**, que se utiliza siempre que sea posible, y **perimeter forwarding**, que se utiliza en las regiones en las que greedy forwarding no se puede aplicar, porque se está en un mínimo local.

## 3.6. CONCLUSIONES DE LOS PROTOCOLOS DE ENCAMINAMIENTO

Como hemos visto disponemos de diferentes protocolos de encaminamiento para las redes malladas inalámbricas. Tenemos disponibles más protocolos que se basan en la topología para calcular las rutas de encaminamiento que en la posición. Esto es así porque los protocolos de encaminamiento basados en la posición son creados específicamente para las redes malladas inalámbricas.

Dentro del grupo de protocolos de encaminamiento basados en la topología, tenemos que la mayoría utilizan el esquema proactivo. Este esquema ofrece una comunicación más rápida debido a que conoce todas las rutas de la red sin necesidad de petición. Presenta un conocimiento de los destinos de la red total y utiliza mensajes continuos para conseguir este objetivo.

B.A.T.M.A.N. es un ejemplo de este tipo, informa al resto de nodos de su existencia a través de mensajes broadcast y crea las rutas al poco tiempo de incluir el nodo.

El inconveniente que presenta los esquemas proactivos es que sobrecargan la red con mensajes de control. Este inconveniente es solventado con los esquemas reactivos que solo se intercambian este tipo de mensajes cuando van a utilizar una ruta. Pero el tiempo de la comunicación aumenta al tener que descubrir la ruta justo antes de enviar el tráfico.

Por ese motivo para elegir el mejor protocolo de encaminamiento hay que estudiar las necesidades de la red y sopesar si necesitamos un protocolo que sea rápido en las comunicaciones aunque la transmisión de mensajes de control sea excesiva, o si por el contrario preferimos una comunicación más lenta pero que no sobrecargue la red con mensajes de control. El dinamismo de la red es crítico para la toma de la decisión, ya que en una red muy dinámica los protocolos proactivos pierden atractivo porque las rutas que descubren están anticuadas para cuando van a usarse.

Por otro lado tenemos los protocolos basados en la posición, como el protocolo de encaminamiento GPSR que presenta robustez en la entrega de paquetes en redes densas, gracias al mecanismo de encaminamiento.

## **CAPITULO 4: DESPLIEGUE DE UNA RED MALLADA BASADA EN EL PROTOCOLO B.A.T.M.A.N.**

## 4.1. INTRODUCCIÓN

En el laboratorio hemos realizado un despliegue de dispositivos para realizar pruebas sobre las redes malladas inalámbricas. Hemos elegido el protocolo B.A.T.M.A.N. para comprender y obtener experiencia sobre los protocolos de encaminamiento de las redes malladas inalámbricas.

Para el montaje de la red mallada inalámbrica disponemos de un ordenador con sistema operativo Linux, 9 routers Linksys WRT54GL y 2 switches.

El ordenador se utiliza para gestionar y controlar los routers, es decir, es el centro de control que nos permite configurar los routers y realizar las pruebas sobre el protocolo B.A.T.M.A.N..

Los 9 routers Linksys WRT54GL son los nodos que se interconectan entre ellos para formar la red mallada inalámbrica. Tienen la capacidad de enviar, recibir y encaminar información. El rango de direcciones IP de los nodos debe pertenecer a la misma subred para que se puedan comunicar entre ellos. Para nuestro proyecto hemos seleccionado el siguiente rango 192.168.1.0/24 y usamos en particular las direcciones 192.168.1.1 a la 192.168.1.10.

Los switches sirven para conectar los nodos al ordenador. Necesitamos 9 puertos Ethernet para cada uno de los nodos y otro puerto adicional que va conectado al ordenador. Como los switches de los que disponemos tienen menos de 10 puertos Ethernet hemos tenido que utilizar 2 switches.

Montamos la red mallada inalámbrica distribuyendo los nodos de la siguiente manera:

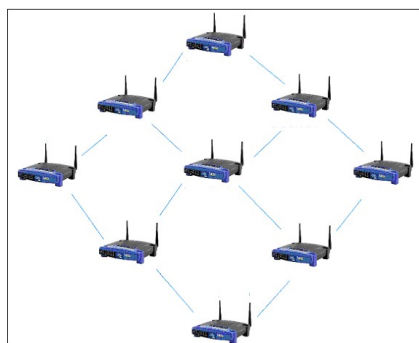


Figura 4.1: Distribución de los routers

Una vez que hemos desplegado los routers y los hemos configurado con el protocolo B.A.T.M.A.N. procedemos a realizar las medidas de rendimiento. Nos centraremos en el retardo del protocolo de encaminamiento cuando se producen cambios en la topología. Además haremos una demostración de acceso a Internet de toda la red mallada cuando uno de los nodos ofrece el acceso.

El objetivo de este proyecto no es hacer pruebas rigurosas de las prestaciones del protocolo, sino obtener experiencia en el funcionamiento. Por tanto con estas pruebas daremos por concluido el proyecto fin de carrera, ya que habremos conseguido el objetivo de experimentar con el protocolo y ver como actúa ante diversas situaciones.

## 4.2. PREPARACIÓN PARA EL DESPLIEGUE DE UNA RED MALLADA BASADA EN B.A.T.M.A.N.

La red mallada inalámbrica soporta redes con topología de infraestructura y permite el acceso a nodos móviles. Es una red muy fiable ya que cada nodo está conectado a varios nodos. Cuando hay que añadir nodos o eliminarlos los vecinos buscan otras rutas. Para todo esto necesitamos de un protocolo de encaminamiento que permita que se transmita la información al destino en el menor número de saltos o en un número de saltos que sea adecuado para cada situación.

Para crear una red mallada inalámbrica necesitamos routers con acceso inalámbrico que funcionen en modo ad-hoc. Para nuestro proyecto utilizaremos los routers Linksys WRT54GL. Además de un punto de control, que en nuestro caso es un ordenador, que se encargará de la administración de los nodos. Para nuestro proyecto utilizaremos 9 routers Linksys WRT54GL.

Cada router Linksys WRT54GL tiene una tarjeta wireless y una o varias tarjetas Ethernet que se utilizan para conectarse con el resto de la red. En los routers Linksys WRT54GL tenemos 4 puertos Ethernet.

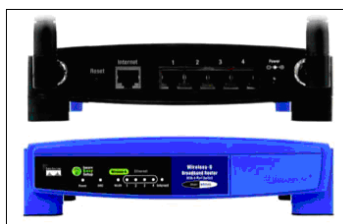


Figura 4.2: Router WRT54GL

En el modo ad hoc las comunicaciones son punto a punto, por lo que si un dispositivo quiere comunicarse con otro que esté fuera de su rango de alcance, envía la información a los dispositivos alcanzables y estos la reenvían hasta que llega a su destino. En este punto entra en juego el protocolo de encaminamiento B.A.T.M.A.N. que se encarga de encontrar el camino desde el origen al destino sin saturar la red. Para configurar el modo ad hoc en los routers debemos acceder a la interfaz web y seleccionar este modo de trabajo. Además debemos asegurarnos de que se encuentren en el mismo canal y ESSID (nombre de la red)[16].

Los dispositivos de fábrica traen un firmware para administrar el router, pero en la mayoría de los casos no son flexibles. Existen varias opciones para dar lugar a configuraciones más avanzadas. Un ejemplo es el OpenWRT que es uno de los más utilizados y el que usaremos en este proyecto. OpenWRT es una distribución Linux que proporciona un sistema totalmente configurable. Nos permite incluir al dispositivo funcionalidades variadas mediante el uso de extensiones (paquetes). OpenWRT ofrece la posibilidad de instalar herramientas como tcpdump para examinar los paquetes que circulan en la red o iptables para configurar el firewall de Linux. También se puede acceder por SSH al sistema Linux para cambiar los parámetros por línea de comandos, que es lo que utilizaremos para instalar el protocolo de encaminamiento B.A.T.M.A.N..

### 4.2.1. CONFIGURACIÓN DE ROUTERS PASO A PASO

Para crear la red mallada inalámbrica debemos configurar todos los routers e incluir el protocolo de encaminamiento B.A.T.M.A.N.. En el Apéndice 1 se explica paso a paso cómo hacerlo.

Básicamente desde el ordenador vamos a configurar todos los routers. Lo primero que debemos hacer es configurar la tarjeta de red Ethernet del ordenador en la misma subred que los routers. Como los routers por defecto vienen con la dirección IP 192.168.1.1/24 elegimos para nuestra tarjeta Ethernet por ejemplo la dirección IP 192.168.1.100.

Una vez configurada la tarjeta Ethernet podemos acceder a los routers Linksys WRT54GL. La primera vez que accedemos al router podemos hacerlo mediante telnet 192.168.1.1 o a través de una página web introduciendo 192.168.1.1.

Como ya hemos comentado antes el firmware original de los routers no es muy flexible y no nos permite instalar nuevos paquetes. Por lo que cambiamos el firmware. En la web <http://downloads.openwrt.org/> podemos descargarnos el firmware que corresponda a nuestra versión del router.

Una vez que hemos cambiado el firmware la función de telnet es desactivada y debe utilizarse ssh.

Durante el cambio de firmware es muy importante no interrumpir el proceso o apagar el router. Si esto sucede el router no actualiza el firmware y deja de funcionar. Existe una manera de recuperarlo, consiste en transferir el firmware mediante tftp. En el Apéndice 1: PASO 6, se explica detalladamente todos los pasos a seguir.

Una vez llegados a este punto vamos a cambiar las direcciones IPs de gestión de los routers y vamos a utilizar el rango 192.168.1.1 al 192.168.1.10.

Vamos a eliminar la petición de password cada vez que accedamos al router. Esto nos facilitará el uso de Scripts y el rápido acceso a los routers. Primero debemos generar la clave y a continuación se transfiere al router. En el Apéndice 1: PASO 7 se explica en detalle el procedimiento.

Una vez que los routers tienen su dirección IP diferente y dentro del rango 192.168.1.1-192.168.1.10, además se ha realizado el cambio de firmware y se ha eliminado la petición de password procedemos a instalar el protocolo B.A.T.M.A.N..

El demonio B.A.T.M.A.N., batmand, lo podemos encontrar en el siguiente enlace: <http://downloads.openwrt.org/kamikaze/8.09/brcm47xx/packages/>

Transferimos el fichero batmand descargado al router mediante scp. Cuando accedemos al router e intentamos instalar batmand mediante opkg nos sale un mensaje de error que nos indica que se debe instalar otros 2 ficheros, una librería y un fichero kmod-tun. Descargamos ambos ficheros del enlace anterior y los instalamos en el router junto a batmand. Todo esto se describe paso a paso en el Apéndice 1, PASO 9.

Ahora que ya hemos instalado batmand en todos los routers procedemos a configurar los parámetros. Debemos configurar la interfaz inalámbrica que vamos a utilizar para el protocolo B.A.T.M.A.N.. Consultamos en el router mediante el comando iwconfig y comprobamos que la interfaz inalámbrica disponible se denomina wl0. Para que la red mallada inalámbrica pueda funcionar debemos configurar todas las interfaces inalámbricas wl0 de todos los routers con el mismo ESSID (nombre de la red), modo y canal. Para nuestro proyecto hemos seleccionado los siguientes parámetros:

- ESSID: **batman-test**
- Mode: **Ad hoc**
- Channel: **1**

Para configurar la interfaz wl0 tenemos dos opciones, desde un terminal mediante el comando, `ifconfig wl0 mode ad-hoc essid batman-test channel 1`, o a través de la web como se indica en el Apéndice 1, PASO 10.

Llegados a este punto vamos a comprobar que los nodos pueden comunicarse. Activamos el batmand en cada router y accedemos a uno de ellos para comprobar que se puede comunicar con el resto. Para activar batmand utilizamos un alias de la interfaz wl0, denominado wl0:bat y le asignamos una dirección IP privada que se encuentre dentro del rango 10.10.0.1-10.10.0.10 con máscara de 16 bits.

Comprobamos que la comunicación mediante ping a las interfaces inalámbricas no funciona y esto es debido a que hay que modificar los Scripts de batmand para que funcionen con nuestros datos. Hay que modificar el Script wireless que se encuentra en la carpeta /etc/config y el Script batmand que se encuentra en la misma carpeta.

En el Script wireless debemos cambiar la opción de red lan por la opción de red wifi. Y en el Script de batmand debemos cambiar la interfaz inalámbrica ath0 por la interfaz wl0.

`vi /etc/config/wireless: option 'network' 'lan' → option 'network' 'wifi'`  
`vi /etc/config/batmand: ath0 → wl0`

Una vez modificados los cambios en los Scripts volvemos a activar batmand y ahora sí funciona el ping a las interfaces inalámbricas de los otros routers.

Por último nos falta instalar los comandos que nos permitirán obtener resultados sobre el funcionamiento del protocolo. Estos comandos son ip route, tcpdump y fping. Ambos comandos los podemos encontrar en el siguiente enlace:

<http://downloads.openwrt.org/kamikaze/8.09/brcm47xx/packages/>

Debemos instalar [ip\\_2.6.25-1\\_mipsel.ipk](#), [tcpdump\\_3.9.8-1.1\\_mipsel.ipk](#) y [fping\\_2.4b2\\_to-ipv6-1\\_mipsel.ipk](#). Para instalar tcpdump necesitamos además la librería [libpcap\\_0.9.8-1\\_mipsel.ipk](#) que se encuentra en el mismo enlace.

En este momento tenemos configurados todos los routers con todo lo necesario para poder realizar las medidas sobre el protocolo B.A.T.M.A.N.

### 4.3. DESPLIEGUE DE UNA RED MALLADA BASADA EN B.A.T.M.A.N.

Vamos a estudiar el funcionamiento del protocolo de encaminamiento B.A.T.M.A.N. con 9 routers Linksys WRT54GL situados de la siguiente manera:

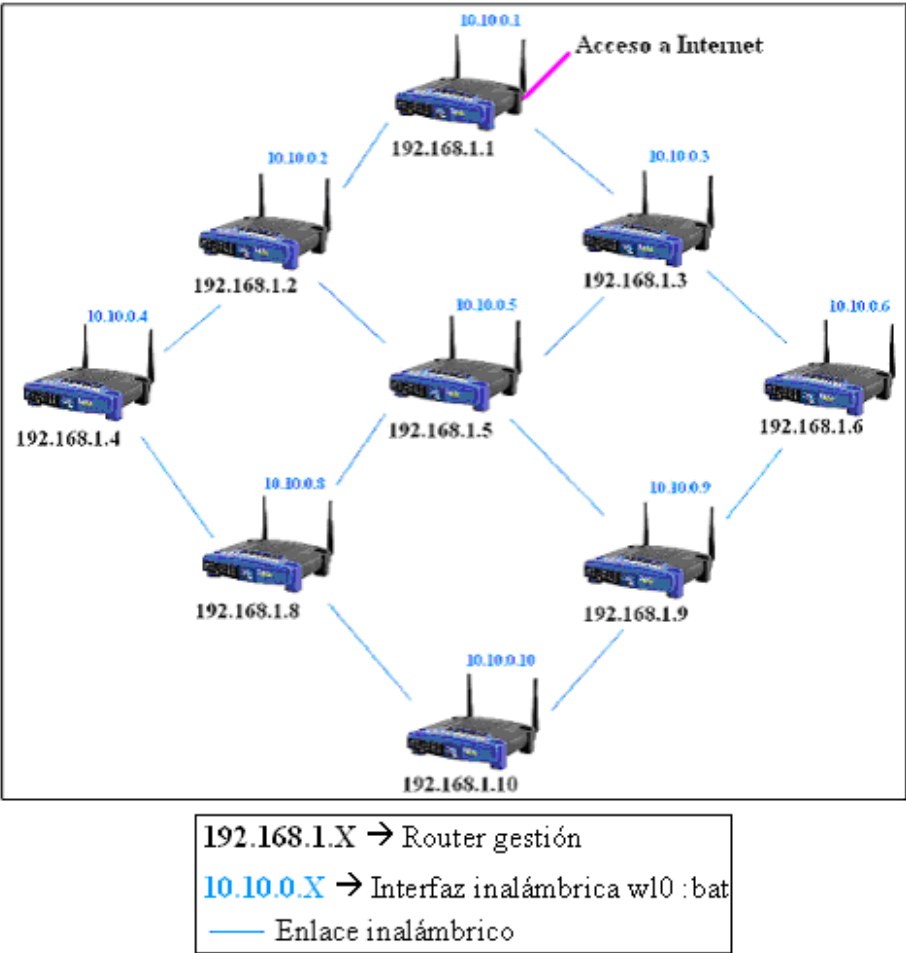


Figura 4.3: Red mallada inalámbrica con 9 routers

Las direcciones **192.168.1.X** hacen referencia a las direcciones IP de gestión de los routers. Y las direcciones **10.10.0.X** a las interfaces inalámbricas wlo:bat que utiliza el protocolo B.A.T.M.A.N..

En una situación real los vecinos de un nodo son aquellos que están dentro del área de cobertura. Como es obvio en un laboratorio todos los routers son vecinos y por tanto debemos utilizar las reglas de firewall para indicar de qué nodos recibe y de qué nodos no. En nuestra red cada nodo tiene como mínimo dos vecinos:

NODO	1	2	3	4	5	6	8	9	10
VECINO	2 3	1 4 5	1 5 6	2 8	2 3 8 9	3 9	4 5 10	5 6 10	8 9

Tabla 4.1: Tabla de nodos vecinos



Utilizamos el comando iptables para definir los nodos vecinos y los nodos que no son vecinos. Se utilizan las direcciones MAC de los nodos para hacer dicha discriminación. No se pueden utilizar las direcciones IP porque se eliminaría toda la información que no tuviera como origen los nodos vecinos. Al utilizar las direcciones MAC se recibe toda la información de los nodos vecinos aunque el nodo que ha originado la información sea otro nodo no vecino. Y eso es exactamente lo que buscamos.

Una vez que ya hemos definido todos los nodos vecinos y los nodos que no son vecinos tal y como se indica en la Figura 4.3, vamos a configurar a cada nodo como gateway o cliente según corresponda.

#### 4.3.1. CONFIGURAMOS LA GATEWAY

El nodo denominado gateway es el que nos permite el acceso a Internet. En nuestro caso es el router 192.168.1.1. Una vez que nos conectamos al router y ejecutamos el comando udhcpd (que ejecuta el módulo que proporciona dhcp), obtenemos una dirección IP dinámica para nuestro router. Desde este momento ya podemos conectarnos a Internet desde la gateway.

Como los nodos de Internet no saben encaminar a las IPs de los clientes de la red mallada inalámbrica, por ser direccionamiento privado, la gateway debe realizar la traducción de las direcciones de la red (funciones NAT: Network Address Translation).

Activamos batmand en el router 192.168.1.1 y anunciamos que permite acceso a Internet con un determinado ancho de banda.

Resumiendo, para configurar el nodo como gateway debemos realizar los siguientes pasos:

**#Accedemos al router:**

ssh [root@192.168.1.1](mailto:root@192.168.1.1)

#Petición de IP

udhcpd -i eth0.1

# Para activar NAT configuramos las reglas MASQUERADE de la interfaz lan del nodo de la gateway:

iptables -t nat -I POSTROUTING 1 -o eth0.1 -j MASQUERADE

iptables -L -t nat -n -vv

#Ejecutamos batmand y anunciamos el ancho de banda que ofrece la gateway.

batmand -g 5000 w10:bat

#### 4.3.2. CONFIGURAMOS LOS CLIENTES

Los nodos clientes son los routers que pertenecen a la red mallada inalámbrica y que necesitan conectarse a los nodos gateway para acceder a Internet. Deben configurarse de modo que detecten los anuncios de los nodos gateways.

Los comandos para configurar a los clientes son los siguientes:

#Accedemos al cliente:  
`ssh root@192.168.1.X`

**#Ejecutamos batmand en el modo que se conecte al enlace que proporcione más calidad en cada momento (aunque para nuestro caso no tenga mucho sentido ya que solo disponemos de una gateway):**

```
batmand -r 3 wl0:bat
```

Vamos a mostrar el resultado de un cliente cuando activamos batmand y detecta una gateway:

```
mbenito@liendre:~/Desktop$ ssh root@192.168.1.10
```

```
BusyBox v1.11.2 (2009-01-06 07:18:07 CET) built-in shell (ash)
Enter 'help' for a list of built-in commands.
```

WIRELESS FREEDOM  
KAMIKAZE (8.09, r14511) -----  
\* 10 oz Vodka      Shake well with ice and strain  
\* 10 oz Triple sec    mixture into 10 shot glasses.  
\* 10 oz lime juice    Salute!

```
root@OpenWrt:~# batmand -r 3 wl0:bat
```

Interface activated: wl0:bat

Using interface wlo:bat with address 10.10.0.10 and broadcast address 10.10.255.255

```
root@OpenWrt:~#
```

#Utilizamos el comando `batmand -c -d 2 -b` para ver el nodo gateway disponible para el acceso a internet.

```
root@OpenWrt:~# batmand -c -d 2 -b
```

```
Gateway (#/255)    Nexthop [outgoingIF], gw_class ... [B.A.T.M.A.N. 0.3.1 rv1206, MainIF/IP: wl0:bat/10.10.0.10, UT: 0d 0h 5m]
=> 10.10.0.1      (108)    10.10.0.8 [ wl0:bat], gw_class 49 - 4MBit/1024KBit, gateway failures: 0
```

```
root@OpenWrt:~#
```

Nos hemos conectado al cliente 10.10.0.10 y vemos que al activar batmand y ejecutar la opción de búsquedas de gateway, ha seleccionado al nodo 10.10.0.1 como gateway.

Automatizamos las configuraciones de los routers en un Script denominado “Conexion a Internet” que se incluye en el Apéndice 2, A2.1. Este Script se conecta a cada uno de los routers y les indica mediante reglas de iptables los nodos que no son vecinos. Además activa batmand como gateway o cliente según corresponda.

### 4.3.3. COMPROBACIONES Y MODIFICACIONES

Primero comprobamos que la red mallada inalámbrica funciona correctamente una vez que hemos definido los vecinos según la Figura 4.3. Para esta comprobación nos conectamos a uno de los routers y lanzamos un ping a todas las interfaces inalámbricas de B.A.T.M.A.N.. En el Apéndice 2, A2.3 incluimos los resultados de realizar ping a todas la interfaces inalámbricas wl0 desde el router 192.168.1.1.



#### **4.4. PRUEBAS DE FUNCIONAMIENTO**

Vamos a realizar varias modificaciones en la topología de la red mallada inalámbrica para ver cómo reacciona la red ante estos cambios. Nos fijaremos en los cambios producidos en las tablas de encaminamiento y en el retardo en recalcular las rutas.

Tomaremos varias medidas del retardo, fijándonos para ello en el tiempo que tarda la red en asumir los cambios. Utilizaremos el comando `fping` que es similar a `ping` pero con la posibilidad de controlar el intervalo de tiempo entre los datagramas.

Las situaciones que vamos a crear en la red mallada para estudiar el retardo son:

- Arranque de la red.
- Enlace caído en un sentido.
- Nodo caído.
- Nodo nuevo

Estas medidas se realizan con el objetivo de obtener experiencia en el protocolo B.A.T.M.A.N., no es un medidor exhaustivo y riguroso de las prestaciones del protocolo.

Además de las medidas del retardo nos vamos a centrar en capturar tráfico dirigido a Internet. Capturaremos tráfico dentro y fuera de la red mallada con el comando `tcpdump`.

##### **4.4.1. RETARDO CUANDO ARRANCA LA RED**

Empezamos las pruebas del retardo cuando arranca la red. Nos referimos al momento que se crea la red y todos los nodos se descubren entre ellos. Para medir este retardo nos vamos a fijar en el tiempo que tarda un router en acceder a otro router.

Vamos a explicarlo brevemente a continuación:

1º. Configuramos uno de los routers, por ejemplo el 192.168.1.1, que pertenece al nodo 10.10.0.1.

2º. En el terminal tenemos preparado el Script “arranque” para ejecutarlo en el momento que se configure el resto de routers.

Este Script “arranque” se conecta al nodo 192.168.1.1 y ejecuta un bucle con el comando `fping` y `date`. Este Script se puede encontrar en el Apéndice 2, A2.5.

Esto nos sirve para saber el instante concreto que se ha ejecutado el Script, que debe coincidir con el momento que la red ya está configurada, y la hora exacta que se alcanza al nodo. Con estos dos valores de tiempo podemos calcular el retardo.

3°. Ahora configuramos el resto de routers, utilizando el Script “Conexión a Internet” eliminando la parte que configura al router 192.168.1.1. Este Script también se encuentra en el Apéndice 2, A2.1.

4° En el momento que termina la configuración de los routers podemos ejecutar el Script “arranque”.

Una vez que tenemos los resultados del Script “arranque” y el comando fping ha conseguido alcanzar a uno de los nodos, por ejemplo el 10.10.0.10. podemos calcular el retardo.

Hemos seleccionado el nodo 10.10.0.1 y el nodo 10.10.0.10 porque la distancia entre ellos es siempre de 4 enlaces, que es lo más lejano posible, y de este modo el resultado del retardo es más real que si elegimos un nodo que se encuentre situado a menos enlaces.

El retardo se obtiene restando el instante en el cual el comando fping ha conseguido conectarse al nodo 10.10.0.10 y el instante que se ejecuto el Script “arranque” que coincide con el momento que todos los routers de la red están configurados. Vamos a explicarlo con la salida del Script “arranque” en una de las pruebas realizadas:

```
mбенито@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:51:21 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:51:22 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:51:23 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:51:24 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:51:25 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
mбенито@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:51:26 UTC 2009
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [0], 84 bytes, 3045 ms (3045 avg, 75% loss)
10.10.0.10 : [1], 84 bytes, 2036 ms (2541 avg, 50% loss)
10.10.0.10 : [2], 84 bytes, 1028 ms (2036 avg, 25% loss)
10.10.0.10 : [3], 84 bytes, 21.3 ms (1532 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 12.2 ms (1228 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 18.5 ms (1027 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 1009 ms (1024 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 5.54 ms (897 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 19.8 ms (799 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 24.4 ms (722 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 5.54/722/3045
Fri Feb 6 02:51:36 UTC 2009
^Cmбенито@liendre:~/Desktop$
```

## RETARDO = 5 segundos

Del resultado obtenido del Script “arranque” vemos que el instante en que el nodo 10.10.0.1 alcanza al nodo 10.10.0.10 es el segundo 26. Y el instante que se ha ejecutado

el Script, que debe coincidir con el momento que todos los nodos se han configurado, es el segundo 21. Por tanto el retardo es de  $26-21=5$  segundos.

Como esta medida es aleatoria ya que el momento en que realizamos el fping no es el momento EXACTO que la red está configurada y puede influir factores como el procesamiento en los nodos, vamos a realizar varias medidas para acercarnos lo más posible al valor real. En el Apéndice 2, A2.5 vienen todas las pruebas realizadas. Aquí vamos a mostrar los resultados del retardo en 30 pruebas:

PRUEBA	1	2	3	4	5	6	7	8	9	10
RETARDO	5	5	7	6	6	4	2	7	6	4

PRUEBA	11	12	13	14	15	16	17	18	19	20
RETARDO	5	4	4	7	5	4	6	11	7	5

PRUEBA	21	22	23	24	25	26	27	28	29	30
RETARDO	11	4	10	11	9	8	5	10	12	13

Tabla 4.2: Retardos cuando arranca la red

Con los resultados obtenidos vamos a calcular la media, sumamos todos los resultados y los dividimos entre 30 que es el número de pruebas y obtenemos que la media del retardo cuando la red arranca es de **6.76 segundos**.

También vamos a calcular el intervalo de confianza que nos proporciona un rango en el cual se encuentra el verdadero valor del retardo con una probabilidad del 95% que es lo que asumiremos para realizar los cálculos. Utilizaremos la función de Excel INTERVALO CONFIANZA e introducimos los siguientes datos:

Alfa = 0.05

Desv\_ estándar = 2,86095394

Tamaño =30

El resultado es: 1,02376041 → **RETARDO = [5.73 , 7.78] Segundos**

Deducimos que el valor real del retardo cuando arranca la red se encuentra en el intervalo de 5.73 y 7.78 segundos con una probabilidad del 0,95. Es un tiempo relativamente pequeño, pero debemos darnos cuenta que estamos desplegando una red de 9 routers, que es muy pequeña. Por tanto podríamos mejorar este resultado disminuyendo el intervalo de tiempo entre los mensajes OGMs para que la red pudiera aprender las rutas con mayor rapidez, a coste de mayor carga de señalización.

#### 4.4.2. RETARDO CUANDO CAE UN ENLACE EN UN SENTIDO

La siguiente prueba que vamos a realizar es medir el tiempo que tarda un nodo en detectar la caída de un enlace en un sentido y modificar la tabla de encaminamiento teniendo en cuenta que ese enlace ya no puede ser usado. Vamos a realizar la medida en una red de 9 routers distribuidos como se muestra en la Figura 4.4.

Las medidas las vamos a realizar en el nodo 10.10.0.10, y el enlace que va a caer es el 10.10.0.10-10.10.0.8. Mostramos la situación en la siguiente Figura:

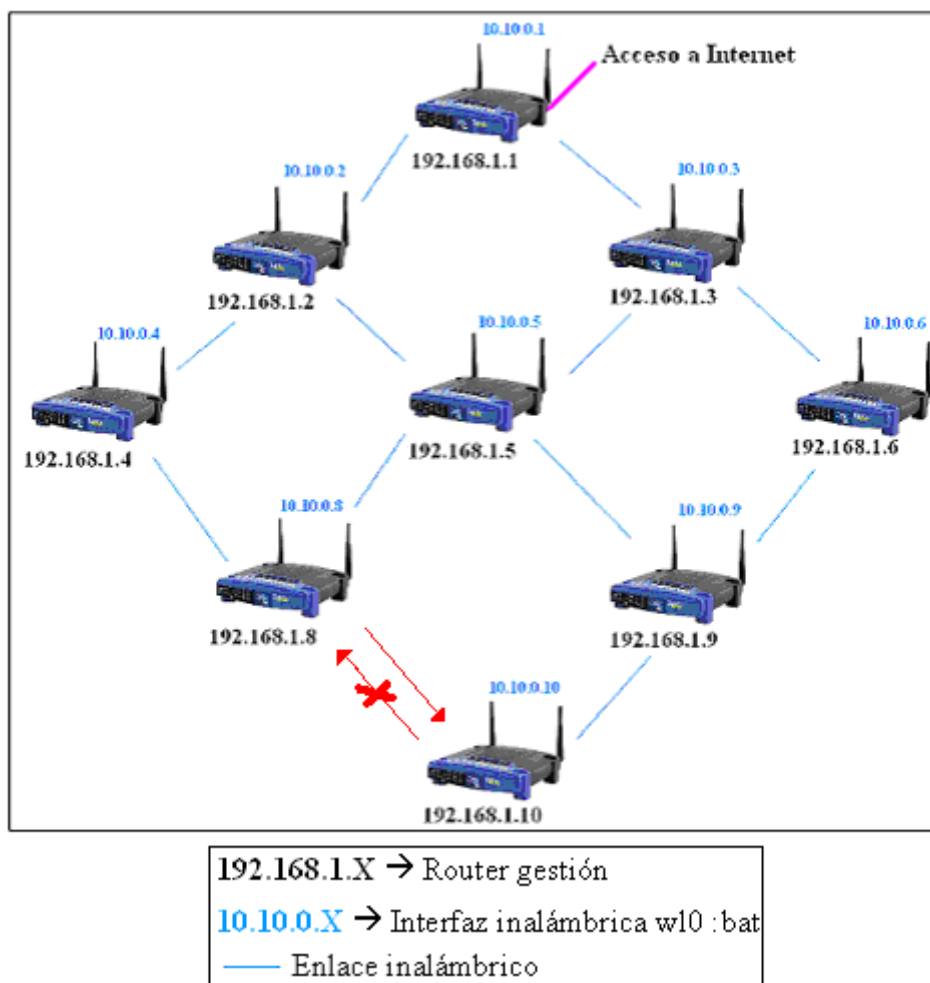


Figura 4.4: Caída del enlace 10.10.0.8 – 10.10.0.10

Lo primero que debemos consultar es la tabla de encaminamiento del router donde vamos a realizar la medida, para nuestro caso el nodo 10.10.0.10. Podemos consultar la tabla de encaminamiento con el siguiente comando:

```
root@OpenWrt:~# ip route list table 66
10.10.0.1 via 10.10.0.8 dev w10 proto static src 10.10.0.10
10.10.0.3 via 10.10.0.8 dev w10 proto static src 10.10.0.10
10.10.0.2 via 10.10.0.8 dev w10 proto static src 10.10.0.10
10.10.0.5 via 10.10.0.8 dev w10 proto static src 10.10.0.10
10.10.0.4 via 10.10.0.8 dev w10 proto static src 10.10.0.10
10.10.0.6 via 10.10.0.8 dev w10 proto static src 10.10.0.10
10.10.0.9 dev w10 proto static scope link src 10.10.0.10
10.10.0.8 dev w10 proto static scope link src 10.10.0.10
```

Del resultado de este comando podemos deducir que el nodo 10.10.0.10 tiene dos vecinos, que son 10.10.0.8 y 10.10.0.9 y que puede alcanzarlos mediante la interfaz inalámbrica w10.

Si forzamos la caída del enlace que une al nodo 10.10.0.8 y 10.10.0.10 en el sentido de 10 a 8 como se muestra en la Figura 4.4, el nodo 10.10.0.10 debe modificar la tabla de

encaminamiento teniendo en cuenta que el nodo 10.10.0.8 ya no es su vecino y por tanto no puede utilizarlo para alcanzar a otros nodos. Además debe buscar si existe alguna alternativa para alcanzar al nodo 10.10.0.8 a través de otro vecino, en este caso el vecino 10.10.0.9.

Si volvemos a consultar la tabla de encaminamiento del nodo 10.10.0.10 vemos que para alcanzar a cualquier nodo utiliza el vecino 10.10.0.9. Y ya no considera el nodo 10.10.0.8 como vecino sino que ha modificado la tabla y ha descubierto que puede alcanzarlo a través del nodo 10.10.0.9. Vemos el resultado de la tabla de encaminamiento una vez que el enlace ha caído:

```
root@OpenWrt:~# ip route list table 66
10.10.0.1 via 10.10.0.9 dev wl0 proto static src 10.10.0.10
10.10.0.3 via 10.10.0.9 dev wl0 proto static src 10.10.0.10
10.10.0.2 via 10.10.0.9 dev wl0 proto static src 10.10.0.10
10.10.0.4 via 10.10.0.9 dev wl0 proto static src 10.10.0.10
10.10.0.6 via 10.10.0.9 dev wl0 proto static src 10.10.0.10
10.10.0.9 dev wl0 proto static scope link src 10.10.0.10
10.10.0.8 via 10.10.0.9 dev wl0 proto static src 10.10.0.10
root@OpenWrt:~#
```

Para obtener el retardo vamos a conectarnos en el router 192.168.1.10 y ejecutamos el comando `fping 10.10.0.8 -i 1 -c 150`. En el momento que se transmite el onceavo datagrama forzamos la caída del enlace en el sentido 10.10.0.10-10.10.0.8. Para conseguir la caída nos conectamos al router 192.168.1.8 y escribimos el comando `iptables -A INPUT -m mac --mac-source 00:1C:10:A4:37:8A -j REJECT`, para indicar al firewall que no acepte mensajes que provengan del la MAC del 10.10.0.10.

A continuación mostramos los comandos y el resultado:

```
root@OpenWrt:~# iptables -A INPUT -m mac --mac-source 00:1C:10:A4:37:8A -j
REJECT
```

```
root@OpenWrt:~# fping 10.10.0.8 -i 1 -c 150
10.10.0.8 : [0], 84 bytes, 7.00 ms (7.00 avg, 0% loss)
10.10.0.8 : [1], 84 bytes, 35.2 ms (21.1 avg, 0% loss)
10.10.0.8 : [2], 84 bytes, 2.65 ms (14.9 avg, 0% loss)
10.10.0.8 : [3], 84 bytes, 9.02 ms (13.4 avg, 0% loss)
10.10.0.8 : [4], 84 bytes, 32.4 ms (17.2 avg, 0% loss)
10.10.0.8 : [5], 84 bytes, 10.2 ms (16.0 avg, 0% loss)
10.10.0.8 : [6], 84 bytes, 2.41 ms (14.1 avg, 0% loss)
10.10.0.8 : [7], 84 bytes, 3.12 ms (12.7 avg, 0% loss)
10.10.0.8 : [8], 84 bytes, 2.74 ms (11.6 avg, 0% loss)
10.10.0.8 : [9], 84 bytes, 30.8 ms (13.5 avg, 0% loss)
10.10.0.8 : [10], 84 bytes, 7.59 ms (13.0 avg, 0% loss)
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
10.10.0.8 : [72], 84 bytes, 12.1 ms (12.9 avg, 83% loss)
10.10.0.8 : [73], 84 bytes, 23.7 ms (13.7 avg, 82% loss)
10.10.0.8 : [74], 84 bytes, 32.1 ms (15.1 avg, 81% loss)
10.10.0.8 : [75], 84 bytes, 16.5 ms (15.1 avg, 80% loss)
10.10.0.8 : [76], 84 bytes, 15.6 ms (15.2 avg, 79% loss)
10.10.0.8 : [77], 84 bytes, 18.5 ms (15.4 avg, 78% loss)
10.10.0.8 : [78], 84 bytes, 33.2 ms (16.4 avg, 77% loss)
10.10.0.8 : [79], 84 bytes, 355 ms (34.2 avg, 76% loss)
10.10.0.8 : [80], 84 bytes, 22.4 ms (33.6 avg, 75% loss)
```



```

10.10.0.8 : [81], 84 bytes, 24.6 ms (33.2 avg, 74% loss)
10.10.0.8 : [82], 84 bytes, 16.8 ms (32.5 avg, 73% loss)
10.10.0.8 : [83], 84 bytes, 20.9 ms (32.0 avg, 72% loss)
10.10.0.8 : [84], 84 bytes, 36.2 ms (32.1 avg, 71% loss)
10.10.0.8 : [85], 84 bytes, 35.5 ms (32.3 avg, 70% loss)
10.10.0.8 : [86], 84 bytes, 27.5 ms (32.1 avg, 70% loss)

```

```

10.10.0.8 : xmt/rcv/%loss = 87/26/70%, min/avg/max = 2.41/32.1/355
root@OpenWrt:~#

```

Analizamos el resultado del comando `ping` y observamos que el tiempo que transcurre desde que forzamos la caída del enlace (segundo 10) hasta que se recupera la comunicación con el nodo 10.10.0.8 (segundo 72) transcurren **62 segundos**. Este sería el retardo pero al igual que antes este valor es aleatorio y por eso realizamos la misma prueba en 30 ocasiones y calculamos la media y el intervalo de confianza. El resto de pruebas se puede consultar en el Apéndice 2, A2.6. Hemos recogido el resultado de las 30 pruebas en la siguiente tabla:

PRUEBA	1	2	3	4	5	6	7	8	9	10
RETARDO	62	49	39	52	64	53	48	53	42	44

PRUEBA	11	12	13	14	15	16	17	18	19	20
RETARDO	42	34	47	40	41	17	47	21	37	28

PRUEBA	21	22	23	24	25	26	27	28	29	30
RETARDO	51	32	35	17	46	43	48	18	44	40

Tabla 4.3: Retardos cuando cae un enlace

Una vez que tenemos todas las pruebas vamos a calcular la media y el intervalo de confianza. La media del retardo nos proporciona un valor de **41.13 Segundos**. Y el intervalo de confianza con los siguientes parámetros:

Alfa = 0.05

Desv\_ estándar = 12,0336692

Tamaño =30

INTERVALO DE CONFIANZA: 4,30611409 → **[36.82 , 45.43] Segundos**

De modo que el valor real del retardo cuando cae un enlace que está siendo utilizado se encuentra en el intervalo de 36.82 y 45.43 segundos. Es un valor elevado y el motivo es que hay que realizar una comprobación con cierta frecuencia para determinar si un enlace bidireccional se ha convertido en un enlace unidireccional. Esta comprobación considera que un enlace sigue siendo bidireccional si el nodo origen recibe del destino uno de los últimos X mensajes OGMs del origen, el parámetro X suele ser por defecto 2. Por tanto, deducimos que un enlace es unidireccional cuando no se recibe los últimos 2 mensajes OGMs transmitidos. Este tiempo podría mejorar si utilizamos un valor del parámetro X más pequeño. [5]

#### 4.4.3. RETARDO CUANDO CAE UN NODO

Ahora vamos a analizar el caso en que uno de los nodos que forman parte de la red mallada inalámbrica cae. En esta situación todas las tablas de encaminamiento que utilicen al nodo caído como mejor vecino deberán modificar sus datos. El tiempo que tardan en recalcular las rutas es el que utilizaremos para calcular el retardo en esta situación.

Vamos a estudiar el retardo en el nodo 10.10.0.10 cuando quiere comunicarse con el nodo 10.10.0.1 y utiliza como mejor vecino el nodo 10.10.0.8, y el nodo 10.10.0.8 deja de funcionar.

Primero nos aseguramos que el nodo 10.10.0.10 utiliza para alcanzar al nodo 10.10.0.1 el vecino 10.10.0.8:

```
root@OpenWrt:~# ip route list table 66
10.10.0.1 via 10.10.0.8 dev wl0 proto static src 10.10.0.10
10.10.0.3 via 10.10.0.8 dev wl0 proto static src 10.10.0.10
10.10.0.2 via 10.10.0.8 dev wl0 proto static src 10.10.0.10
10.10.0.5 via 10.10.0.8 dev wl0 proto static src 10.10.0.10
10.10.0.4 via 10.10.0.8 dev wl0 proto static src 10.10.0.10
10.10.0.6 via 10.10.0.9 dev wl0 proto static src 10.10.0.10
10.10.0.9 dev wl0 proto static scope link src 10.10.0.10
10.10.0.8 dev wl0 proto static scope link src 10.10.0.10
```

Vemos que si utiliza el vecino 10.10.0.8 para comunicarse con el nodo 10.10.0.1 por lo que procedemos a forzar la caída del nodo. La situación que tenemos es la siguiente:

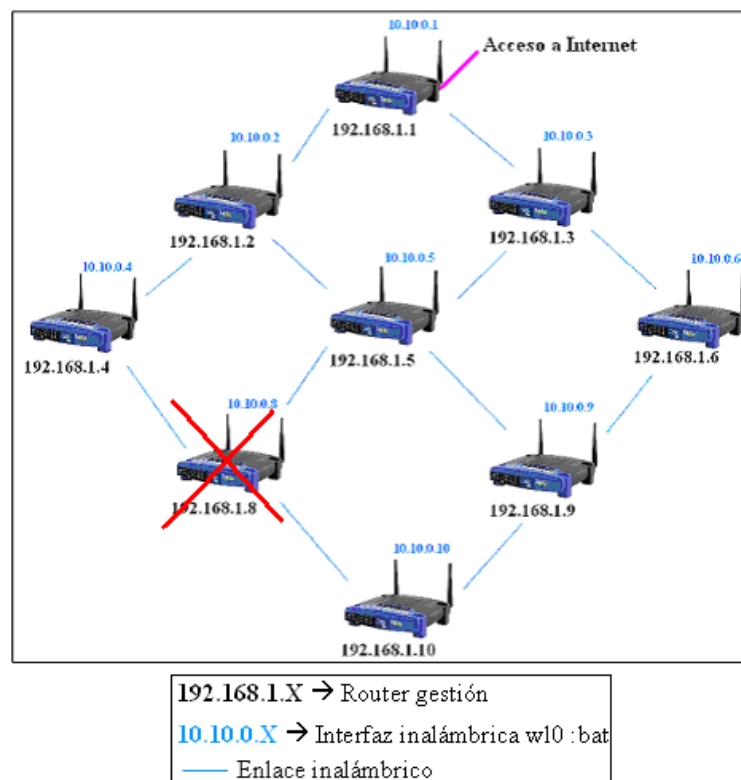


Figura 4.5: Caída del nodo 10.10.0.10



<b>PRUEBA</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>30</b>
<b>RETARDO</b>	<b>15</b>	<b>13</b>	<b>14</b>	<b>17</b>	<b>17</b>	<b>11</b>	<b>24</b>	<b>24</b>	<b>17</b>	<b>18</b>

Tabla 4.4: Retardos cuando cae un nodo

Calculamos la media de todos los resultados obtenidos en las 30 pruebas, y obtenemos que el retardo medio es de **15.36 Segundos**.

También vamos a calcular el intervalo de confianza, los datos que vamos a utilizar son los siguientes:

Alfa = 0.05

Desv\_estándar = 3,36803773

Tamaño = 30

INTERVALO DE CONFIANZA: 1,2052146  $\rightarrow$  **RETARDO = [14.15 , 16.56]**

Como podemos observar el retardo cuando cae un nodo tiene un valor entre 14.15 y 16.56 Segundos. Este resultado es mayor que el retardo del arranque, un poco más del doble. Y es menor que el retardo cuando cae un enlace, casi 3 veces menos. Esta diferencia con el caso anterior se debe a que ahora la caída de un nodo se detecta más rápidamente porque antes el enlace seguía disponible a nivel 2 aunque no fuese posible la comunicación, ya que la teníamos filtrada.

#### 4.4.4. RETARDO CUANDO INTRODUCIMOS UN NODO NUEVO

Para esta situación vamos a partir de una red de 8 routers y vamos a medir el retardo cuando se incluye un nuevo nodo. Las medidas las realizaremos en el nodo que se encuentre más distante al nuevo nodo. La distribución de los 8 routers es como se muestra en la siguiente figura:

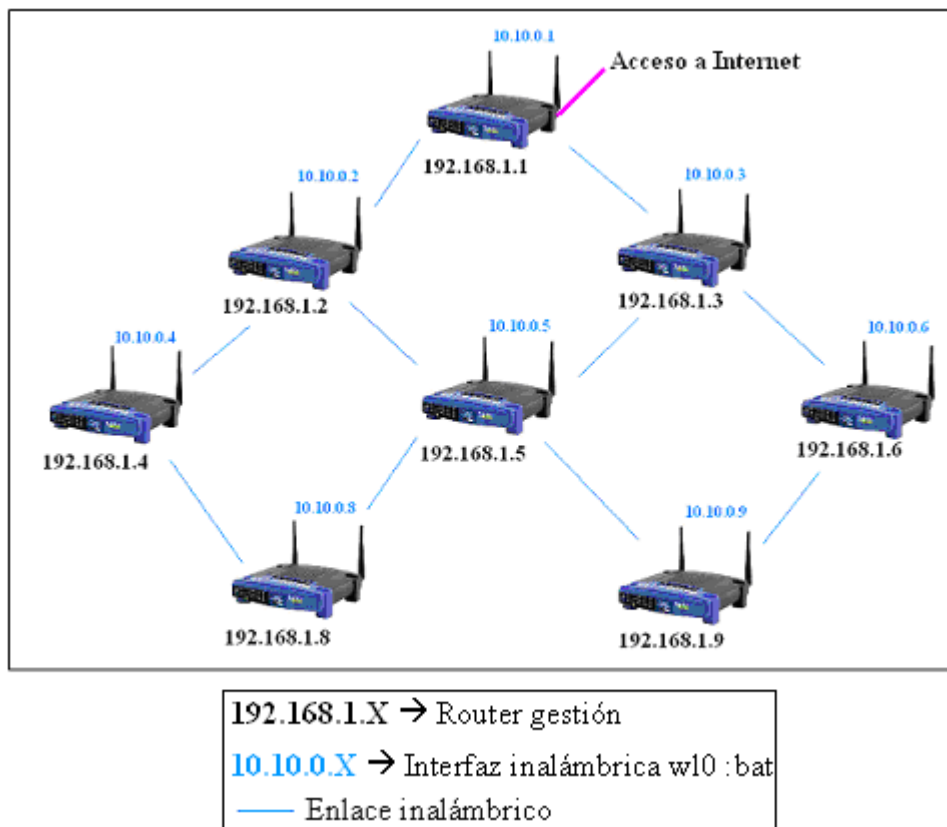


Figura 4.6: Red mallada de 8 routers

Una vez que se haya configurado la red de 8 routers como se indica en la Figura 4.6, vamos a introducir un nuevo nodo 10.10.0.10 que se situará entre los nodos 10.10.0.8 y 10.10.0.9.

La prueba consistirá en ver el tiempo que tarda el nodo 10.10.0.1, que es el que más lejano al nodo nuevo, en alcanzar al nodo 10.10.0.10.

Los pasos a seguir son los siguientes:

- Configuramos la red de 8 routers con el Script “Conexión a internet” (que se encuentra en el Apéndice 2, A2.1) pero eliminando la configuración del nodo 10.10.0.10.
- Nos conectamos a un terminal y tenemos preparados el Script denominado “arranque” para ejecutarlo en el momento que introducimos el nodo y activamos batmand. Este Script se conecta al router 192.168.1.1 y ejecuta el comando fping y date para obtener el instante que el nodo se ha introducido y el instante que se puede acceder a él. El Script “arranque” se encuentra en el Apéndice 2, A2.5.
- Conectamos el nuevo router 192.168.1.10 y activamos batmand:

```
mбенито@лиендре:~$ ssh root@192.168.1.10 '/etc/init.d/firewall stop ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:36:43 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:A3:A7 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:36:67 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:98:4C -j REJECT ; iptables -A INPUT -m mac --mac-source
```

```
00:1C:10:A4:2C:59 -j ACCEPT ; iptables -A INPUT -m mac --mac-source
00:1C:10:9A:72:8D -j ACCEPT ; killall batmand ; /etc/init.d/batmand stop ; ifconfig
wl0:bat 10.10.0.10 netmask 255.255.0.0 broadcast 10.10.255.255 ; batmand -r 3 wl0:bat '
```

- Ahora ejecutamos el Script “arranque” en el terminal y vemos que el resultado es el siguiente:

```
mбенито@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:50:11 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:12 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:13 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:14 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:15 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:16 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:17 UTC 2009
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [0], 84 bytes, 49.9 ms (49.9 avg, 0% loss)
10.10.0.10 : [1], 84 bytes, 14.7 ms (32.3 avg, 0% loss)
10.10.0.10 : [2], 84 bytes, 21.4 ms (28.7 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 40.2 ms (31.6 avg, 20% loss)
10.10.0.10 : [5], 84 bytes, 7.66 ms (26.8 avg, 16% loss)
10.10.0.10 : [6], 84 bytes, 11.8 ms (24.3 avg, 14% loss)
10.10.0.10 : [7], 84 bytes, 17.4 ms (23.3 avg, 12% loss)
10.10.0.10 : [8], 84 bytes, 10.3 ms (21.7 avg, 11% loss)
10.10.0.10 : [9], 84 bytes, 112 ms (31.7 avg, 10% loss)

10.10.0.10 : xmt/rcv/%loss = 10/9/10%, min/avg/max = 7.66/31.7/112
Fri Feb 6 02:50:27 UTC 2009
^Cmбенито@liendre:~/Desktop$
```

## RETARDO = 6 SEGUNDOS

Del resultado del Script “arranque” podemos ver que a las 2:50:11 se ejecuto el Script y es el instante en el que el nuevo nodo 10.10.0.10 es configurado. El comando fping no muestra resultados hasta las 2:50:17. Por tanto el tiempo que ha transcurrido son **6 segundos** y este sería el retardo del nodo 10.10.0.1 en alcanzar al nodo 10.10.0.10.

Para que el resultado del retardo cuando introducimos un nuevo nodo sea más fiable tomamos más medidas en esta misma situación. Los resultados del resto de pruebas se pueden ver en el Apéndice 2, A2.8. Vamos a reunir todos los resultados en una tabla:

PRUEBA	1	2	3	4	5	6	7	8	9	10
RETARDO	6	8	8	5	3	4	4	9	4	5

PRUEBA	11	12	13	14	15	16	17	18	19	20
RETARDO	4	5	3	5	5	7	5	7	6	5

PRUEBA	21	22	23	24	25	26	27	28	29	30
RETARDO	7	4	7	2	3	4	5	4	4	6

Tabla 4.5: Retardos cuando introducimos un nodo nuevo

Como en los anteriores casos vamos a calcular la media y el intervalo de confianza. El resultado de la media es de **5.13 Segundos**. Y el resultado del intervalo de confianza con los siguientes datos es:

Alfa = 0.05

Desv\_estándar = 1,67606545

Tamaño =30

INTERVALO DE CONFIANZA: 0,5997613 → **RETARDO = [4.53 , 5.73]**

Por tanto el retardo cuando introducimos un nuevo nodo en una red formada por 8 routers oscila entre los 4.53 y 5.73 Segundos. Es el retardo más pequeño que hemos obtenido de todas las situaciones en que hemos realizado medidas. Incluso más pequeño que cuando la red arranca. Esto es así porque en esta situación solo hay que incluir en la tabla de encaminamiento un nuevo nodo, mientras que cuando arranca la red hay que incluir a 8 nodos.

#### 4.4.5. ANÁLISIS DE TRÁFICO DIRIGIDO A INTERNET

Cuando los nodos clientes quieren navegar por Internet, seleccionan el nodo gateway de la red mallada inalámbrica y se crea un túnel punto a punto entre el nodo cliente y el nodo gateway. Cuando disponemos de más de un nodo gateway, el nodo cliente es el encargado de decidir qué gateway utilizar.

El túnel creado es denominado gate0 y el rango de direcciones IP es 169.254.0.0/16.

El paquete original que crea el nodo cliente es encapsulado como carga UDP y se envía a través del túnel gate0 al nodo gateway. Las direcciones origen y destino del paquete original no cambian. La dirección origen es la dirección IP gate0 del nodo cliente. Y la dirección destino es la dirección IP de Internet.

La cabecera IP exterior del paquete encapsulado contiene como dirección origen el nodo cliente y como dirección destino el nodo gateway.

Si el nodo cliente no es vecino del nodo gateway, los nodos intermedios encaminan los paquetes de un punto a otro fijándose en el mejor vecino de salto que contengan sus tablas de encaminamiento.

Todo lo que hemos explicado hasta aquí se refiere a los paquetes que se encuentran dentro de la red mallada inalámbrica. Si analizamos los paquetes que se encuentran fuera de la red mallada inalámbrica, vemos que la dirección IP que se utiliza es la dirección pública del nodo gateway que ofrece acceso a Internet. En la captura de tráfico realizada fuera de la red podemos ver esto que se ha explicado.

El puerto 4305 ha sido asignado por IANA para uso exclusivo del protocolo B.A.T.M.A.N. Y el puerto 4306 es definido por IANA como pingh1 Hellgate London.

Para analizar los paquetes capturados vamos a explicar el formato del datagrama IP y sus campos. Como ya sabemos, el datagrama IP es la unidad básica de transferencia de datos entre el origen y el destino. El formato es el siguiente:

0	10								20								30														
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
VERS				HLEN				Tipo de servicio								Longitud total															
Identificación												Bandrs				Desplazaiento de fragmento															
TTL						Protocolo								CRC cabecera																	
Dirección IP origen																															
Dirección IP destino																															
Datos																															
...																															

Figura 4.7: Datagrama IP

- Version: Versión del protocolo IP que se utilizó para crear el datagrama.
- HLEN: Longitud de la cabecera.
- Tipo de servicio: Indica la calidad del servicio.
- Longitud total: Longitud total del datagrama.
- Identificación: Número de secuencia.
- Banderas o flags: Indica si el datagrama se puede fragmentar.
- Desplazamiento de fragmento: Lugar donde insertar el fragmento actual en el datagrama completo.
- TTL: Time To Live, tiempo de vida. Número máximo de segundos que puede estar el datagrama en la red.
- Protocolo: Protocolo utilizado en el campo de datos. 1 ICMP, 2 IGMP, 6 TCP y 17 UDP.
- Dirección origen: Dirección IP origen.
- Dirección destino: Dirección IP destino.
- Datos: Contiene los datos a enviar. Este campo comienza con la cabecera del siguiente nivel TCP o UDP. La cabecera UDP indica el puerto origen y destino, la longitud del mensaje y la suma de verificación:

+	Bits 0 - 15	16 - 31
0	Puerto origen	Puerto destino
32	Longitud del Mensaje	Suma de verificación
64	Datos	

Figura 4.8: Cabecera UDP

Ahora vamos a realizar una captura de tráfico dirigido hacia Internet (google.es) dentro y fuera de la red mallada. Al igual que antes disponemos de una red de 9 routers dispuestos como se muestra en la siguiente Figura y realizamos un ping desde el nodo 10.10.0.10 a la dirección google.es:



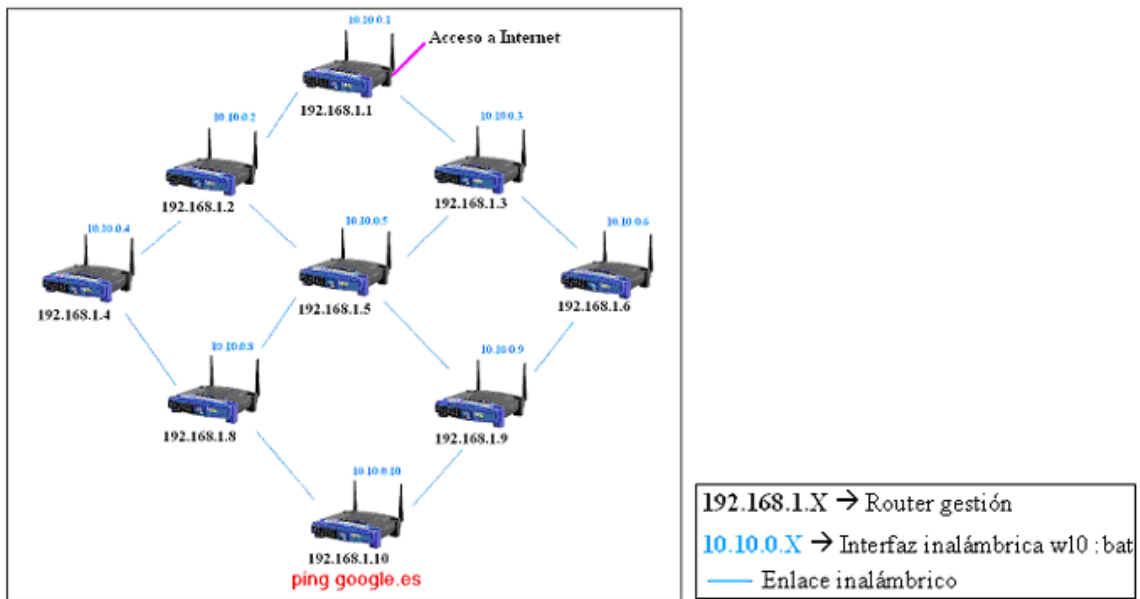


Figura 4.9: Acceso a google.es

Primero configuramos la red indicando el nodo 10.10.0.1 como nodo gateway y el resto como nodos clientes. Para esto ejecutamos el Script de configuración guardado en “Conexion a internet”, que se encuentra en el Apéndice 2, A2.1.

```
mбенито@liendre:~$ sh Conexion\ a\ internet
```

Ahora nos conectamos al router 192.168.1.1 y realizamos una captura en la interfaz inalámbrica wl0 para capturar el tráfico dentro de la red mallada inalámbrica. Y una captura en la interfaz eth0.1 para analizar el tráfico que se cursa fuera de la red mallada inalámbrica.

En el momento que se captura el tráfico en el nodo 10.10.0.1 mediante el comando tcpdump lanzamos un ping a Internet desde el nodo 10.10.0.10:

```
mбенито@liendre:~$ ssh root@192.168.1.10
```

```
BusyBox v1.11.2 (2009-01-06 07:18:07 CET) built-in shell (ash)
Enter 'help' for a list of built-in commands.
```

```

|_| .-----|_|_| .-----|_|_| | | | | | | | | |
|_| - || _| - || || || || ||
|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
|_| WIRELESS FREEDOM
KAMIKAZE (8.09, r14511) -----
* 10 oz Vodka    Shake well with ice and strain
* 10 oz Triple sec  mixture into 10 shot glasses.
* 10 oz lime juice Salute!
-----
```

```
root@OpenWrt:~# ping google.es
PING google.es (209.85.229.104): 56 data bytes
64 bytes from 209.85.229.104: seq=0 ttl=50 time=46.438 ms
64 bytes from 209.85.229.104: seq=1 ttl=50 time=155.321 ms
64 bytes from 209.85.229.104: seq=2 ttl=50 time=35.803 ms
```

```
--- google.es ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 35.803/79.187/155.321 ms
root@OpenWrt:~#
```

## Captura de tráfico dentro de la red mallada:

Nos conectamos al router 192.168.1.1 y ejecutamos el comando `tcpdump -i wl0 port 4306 -X > dentro` en el momento que el nodo 10.10.0.10 se conecta a Internet, [ping google.es](#). El comando tcpdump captura el tráfico que circula por la interfaz inalámbrica wl0 y tiene como origen o destino el puerto 4306. Este puerto es el que utiliza la gateway para enviar el tráfico encapsulado en paquetes UDP. El fichero “dentro” es el que contiene el resultado de la captura del comando tcpdump y que se puede consultar en el Apéndice 2, A2.9.1.

Las direcciones IPs utilizadas son:

**10.10.0.1** → GATEWAY → **0a 0a 00 01**

**10.10.0.10** → CLIENTE → **0a 0a 00 0a**

169.254.0.7 → gate0 **10.10.0.10** → **a9 fe 00 07**

163.117.139.253 → DNS → **a3 75 8b fd**

209.85.229.104 → google.es → **d1 55 e5 68**

Vamos a resaltar cada dirección para analizar el tráfico de paquetes, solo utilizaremos unos paquetes de ejemplo y el resto se puede consultar en el Apéndice 2, A2.9.1:

```
root@OpenWrt:~# cat dentro
03:00:31.798298 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 56
  0x0000: 4500 0054 0000 4000 3d11 297b 0a0a 000a E..T..(.=.){....
  0x0010: 0a0a 0001 10d2 10d2 0040 7585 0145 0000 .....@u..E..
  0x0020: 37ec f740 0040 1174 46a9 fe00 07a3 758b 7..@..@.tF.....u.
  0x0030: fd08 0000 3500 2358 3f00 0201 0000 0100 ....5#X?.....
  0x0040: 0000 0000 0006 676f 6f67 6c65 0265 7300 .....google.es.
  0x0050: 001c ..
03:00:31.800605 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 116
  0x0000: 4500 0090 0000 4000 4011 263f 0a0a 0001 E.....@..@.&?...
  0x0010: 0a0a 000a 10d2 10d2 007c b10d 0145 0000 .....].E..
  0x0020: 7300 0040 003d 1164 02a3 758b fda9 fe00 s..@.=.d.u....
  0x0030: 0700 3508 0000 5f91 5d00 0281 8000 0100 ..5...]......
  0x0040: 0000 0100 0006 676f 6f67 6c65 0265 7300 .....google.es.
  0x0050: 001c ..
03:00:32.004357 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
  0x0000: 4500 0071 0000 4000 3d11 295e 0a0a 000a E..q..@.=.)^....
  0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
  0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@..@.....U.
  0x0030: 0808 0020 86d1 0200 0048 43be 3300 0000 h.....HC.3...
  0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0x0050: 0000 ..
03:00:32.071548 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
  0x0000: 4500 0071 0000 4000 4011 265e 0a0a 0001 E..q..@..@.&^....
  0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
  0x0020: 54a0 f200 0032 0186 f3d1 55e5 68a9 fe00 T.....2...U.h...
  0x0030: 0700 0028 86d1 0200 0048 43be 3300 0000 ...(.HC.3...
  0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0x0050: 0000 ..
```

Como podemos observar en la captura de tráfico todos los mensajes tienen origen o destino los nodos 10.10.0.1 y 10.10.0.10 (nodo gateway y nodo cliente).

En los primeros mensajes se realiza la petición al DNS. La dirección origen es la interfaz gate0 **a9 fe 00 07** del nodo 10.10.0.10, y la dirección del DNS es **a3 75 8b fd**.

Una vez que tenemos la respuesta del DNS comienza la comunicación con la dirección google.es, **d1 55e5 68**.

Para comprender mejor los valores del datagrama vamos a estudiar el último datagrama campo a campo:

```
4500 0071 0000 4000 3d11 295e 0a0a 0001 E..q..(.=.)^....
0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
54a0 f400 0032 0186 f1d1 55e5 68a9 fe00 T...2....U.h...
0700 00d4 09d1 0200 027e bddc 3300 0000 .....~.3...
0000 0000 0000 0000 0000 0000 0000 0000 .....
0000
```

010234567890123356789012345678901																															
VERS				HLEN				Tipo de servicio				Longitud total																			
Identificación								Bandrs				Desplazaiento de fragmento																			
TTL				Protocolo				CRC cabecera																							
Dirección IP origen																															
Dirección IP destino																															
Datos																															
...																															

010234567890123356789012345678901																															
4				5				00				0071																			
0000								4				000																			
3d				11				295e																							
0a0a 0001																															
0a0a 000a																															
10d2 10d2																															
005d c871																															
...																															
f1d1 55e5																															
68a9 fe00																															
0700 00d4																															
...																															

Figura 4.10: Comparación de datagramas IP

- Version: 4 corresponde con IPv4.
- HLEN: 5 bytes.
- Tipo de servicio: 0 calidad baja.
- Longitud total: 113 bytes.
- Identificación: Número de secuencia 0000.
- Banderas o flags: No fragmentar.
- Desplazamiento de fragmento: 000.
- TTL: Time To Live, tiempo de vida. 61 segundos.
- Protocolo: 17 UDP.
- Dirección origen: 10.10.0.1.
- Dirección destino: 10.10.0.10.
- Datos: Cabecera UDP que contiene el puerto origen, destino, longitud del mensaje, suma de verificación y los datos.
  - o Puerto origen: 4306.
  - o Puerto destino: 4306.
  - o En los datos encontramos la dirección origen y destino del paquete original:
    - Dirección origen: d155 e568 google.es.
    - Dirección destino: a9fe 0007 gate0 del cliente 10.10.0.10.

Otra prueba realizada sobre este mismo escenario pero capturando tráfico en el nodo 10.10.0.8 en la interfaz wl0 sin especificar el puerto se puede encontrar en el Apéndice 2, A2.9.2. Vemos que en el puerto 4305 se encuentran los paquetes que van dirigidos a la red mallada inalámbrica, dirección 10.10.255.255, y en el puerto 4306 los paquetes dirigidos fuera de la red mallada inalámbrica. Vamos a mostrar sólo unos paquetes de ejemplo para ver lo que explicamos:

Paquete que utiliza el Puerto 4305 y que tiene como destino la dirección 10.10.255.255:

```
03:09:50.825080 IP 10.10.0.9.4305 > 10.10.255.255.4305 : UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E.....@.@.&.....
0x0010: 0a0a ffff 10d1 10d1 001a 02d0 0500 3000 .....0.
0x0020: 033c 10d2 0a0a 0003 0a0a 0006 6a00 .<.....j.
```

10.10.0.9 (0a0a 0009) → Nodo que forma parte de la red mallada inalámbrica.  
 10.10.255.255 (0a0a ffff) → Dirección de broadcast.

Paquete que utiliza el Puerto 4306 y que tiene como destino Google.es:

```
03:09:48.927448 IP 10.10.0.10.4306 > 10.10.0.1.4306 : UDP, length 85
0x0000: 4500 0071 0000 4000 3e11 285e 0a0a 000a E..q..@.>.(^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@.@.....U.
0x0030: 6808 00ee 16d8 0200 01b5 8b7c 5900 0000 h.....|Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000
```

10.10.0.10 (0a0a 000a) → Nodo cliente.  
 10.10.0.1 (0a0a 0001) → Nodo Gateway.

169.254.0.7 (a9 fe00 07) → Es la interfaz gate0 del cliente 10.10.0.10.  
 209.85.229.104 (d1 55e5 68) → Es la dirección de Google.es 209.85.229.104

Con este ejemplo queremos mostrar el uso de los puertos 4305 y 4306. Por tanto, el puerto 4305 se utiliza para los mensajes OGM, que informan a la red de la existencia del nodo. Y el puerto 4306 para el túnel UDP de los paquetes que se dirigen fuera de la red mallada inalámbrica.

### Captura de tráfico fuera de la red mallada:

Partimos de la misma situación que se muestra en la Figura 4.9. Y ahora vamos a mostrar los paquetes que atraviesan la interfaz eth0.1 que es la que se conecta a Internet. El objetivo de esta captura es analizar el formato de los paquetes fuera de la red mallada inalámbrica. Para hacer esto realizamos una captura en la interfaz eth0.1 cuando el cliente 10.10.0.10 accede a Internet. El comando que tenemos que ejecutar en la gateway 10.10.0.1 es el siguiente:

```
tcpdump -i eth0.1 -X > fuera
```

El fichero “fuera” es el que contiene el resultado de la captura. Las direcciones IPs utilizadas en la captura son las siguientes:

163.117.140.216 → eth0.1 → a375 8cd8  
 163.117.139.253 → DNS varpa.it.uc3m.es → a375 8bfd  
 209.85.229.104 → google.es / ww-in-fl04.1e100.net → d155 e568

Vamos a mostrar una muestra de los paquetes capturados y el resto de la captura se puede consultar en el fichero “fuera” que se puede consultar en el Apéndice 2, A2.10.

```
03:37:36.692447 IP 163.117.140.216.2048 > varpa.it.uc3m.es. 53: 2+ AAAA? google.es. (27)
0x0000: 4500 0037 5212 4000 3f11 89e3 a375 8cd8 E..7R.@.?....u..
0x0010: a375 8bfd 0800 0035 0023 d1f6 0002 0100 .u.....5#.....
0x0020: 0001 0000 0000 0000 0667 6f6f 676c 6502 .....google.
```

0x0030: 6573 0000 1c00 01 es.....

03:37:36.738638 IP **varpa.it.uc3m.es**. 53 > **163.117.140.216**.2048: 5 3/4/4 A **ww-in-f104.1e100.net**,[[domain]  
0x0000: 4500 00f9 0000 4000 3e11 dc33 **a375 8bfd** E.....@.>..3.u..  
0x0010: **a375 8cd8** 0035 0800 00e5 42fb 0005 8180 .u...5...B.....  
0x0020: 0001 0003 0004 0004 0667 6f6f 676c 6502 .....google.  
0x0030: 6573 0000 0100 01c0 0c00 0100 0100 0005 es.....  
0x0040: c700 04d1 55e5 68c0 0c00 0100 0100 0005 ....U.h.....  
0x0050: c700 ..

03:37:37.761616 IP **163.117.140.216** > **ww-in-f104.1e100.net**: ICMP echo request, id 57346, seq 1, length 64  
0x0000: 4500 0054 7321 0000 3f01 549d **a375 8cd8** E..T..@..?.T..u..  
0x0010: **d155 e568** 0800 c8a3 e002 0001 e69f 68b8 .U.h.....h.  
0x0020: 0000 0000 0000 0000 0000 0000 0000 0000 .....  
0x0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....  
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....  
0x0050: 0000 ..

03:37:37.785564 IP **ww-in-f104.1e100.net** > **163.117.140.216**: ICMP echo reply, id 57346, seq 1, length 64  
0x0000: 4500 0054 7321 0000 3201 2e7c **d155 e568** E..Ts!..2..|U.h  
0x0010: **a375 8cd8** 0000 d0a3 e002 0001 e69f 68b8 .u.....h.  
0x0020: 0000 0000 0000 0000 0000 0000 0000 0000 .....  
0x0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....  
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....  
0x0050: 0000 ..

Del resultado de la captura podemos deducir que la IP que se utiliza cuando salimos fuera de la red mallada inalámbrica es **163.117.140.216**, que pertenece a la dirección IP de la interfaz eth0.1.

En los primeros paquetes de la captura se consulta al servidor DNS la dirección de google.es. Esto se puede observar en el primer paquete que tiene como origen la interfaz eth0.1 (**163.117.140.216**) y como destino el servidor DNS (**varpa.it.uc3m.es**). En el segundo paquete recibimos la respuesta del DNS a la pregunta de la dirección google.es. En este paquete la dirección origen es el servidor DNS (**varpa.it.uc3m.es**) y el destino la interfaz eth0.1 (**163.117.140.216**) y la respuesta es la dirección google.es (**ww-in-f104.1e100.net**).

Una vez que se ha obtenido la dirección de google.es, empieza la comunicación directa entre la gateway (**163.117.140.216**) y google.es (**ww-in-f104.1e100.net**).

#### 4.5. VALORACIÓN DE LOS RESULTADOS

En este capítulo hemos presentado las experiencias obtenidas con el protocolo B.A.T.M.A.N. en el laboratorio. A continuación vamos a reunir los resultados obtenidos de los retardos para analizarlos.

SITUACION DE LA TOPOLOGIA	MEDIA DEL RETARDO	INTERVALO DE CONFIANZA (Segundos)
<b>ARRANQUE DE LA RED</b>	<b>6.76 Segundos</b>	<b>[5.73 , 7.78]</b>
<b>CAIDA DE UN ENLACE</b>	<b>41.13 Segundos</b>	<b>[36.82 , 45.43]</b>
<b>CAIDA DE UN NODO</b>	<b>15.36 Segundo</b>	<b>[14.15 , 16.56]</b>
<b>INSERCIÓN DE UN NODO NUEVO</b>	<b>5.13 Segundos</b>	<b>[4.53 , 5.73]</b>

Tabla 4.6: Retardos en las distintas situaciones

Como ya hemos mencionado antes los valores de tiempo son elevados teniendo en cuenta que la red mallada inalámbrica está formada por 9 routers. El objetivo de este proyecto no es presentar al protocolo B.A.T.M.A.N. como mejor opción para las redes malladas inalámbricas sino obtener experiencia en este tipo de protocolo de encaminamiento y obtener resultados para comparar con otros protocolos existentes.

Los valores de retardo más pequeños corresponden con la situación de arranque de la red y la inserción de un nodo nuevo. Estos valores podrían reducirse aún más si aumentamos el número de mensajes OGMs, porque así la red se informaría de los cambios topológicos con mayor rapidez. Esta misma acción también mejoraría los retardos en la situación de caída de nodo o de enlace.

Los resultados nos muestran que la red actúa con mayor rapidez cuando tiene que incluir nuevos datos en las tablas de encaminamiento que cuando tiene que modificarlas. Esto es así porque se detecta antes la inserción de nuevos nodos mediante los mensajes OGMs que la caída de un nodo o enlace que tiene que detectarse y recalcular rutas.

En el apartado anterior hemos visto que es posible el acceso a Internet de toda la red mallada inalámbrica cuando tenemos un nodo que nos permite el acceso. Esto es una gran ventaja ya que toda la red tiene acceso disponible a Internet y a otras redes.

Hemos analizado el tráfico dentro y fuera de la red mallada inalámbrica cuando un nodo cliente accede a Internet. El resultado nos ha demostrado el uso de túneles UDP entre el nodo cliente y el nodo gateway para conseguir el acceso a Internet.

Los paquetes que se intercambian dentro de la red mallada inalámbrica y que utilizan los túneles UDP denominados gate0, tienen como cabecera IP externa la dirección del nodo cliente y del nodo gateway. Si el paquete es una petición el nodo cliente es el origen y el nodo gateway es el destino. Y si el paquete es una respuesta el nodo gateway es el origen y el nodo cliente es el destino. Dentro del paquete se indica la dirección de la interfaz gate0 del cliente y la dirección de Internet. Y al igual que antes, si el paquete es una petición la dirección gate0 del cliente es el origen y la dirección de Internet es el destino. Y si se trata de un paquete respuesta la dirección origen es la de Internet y la dirección destino es la dirección de la interfaz gate0 del cliente.

Los paquetes generados dentro de la red mallada inalámbrica cuando salen fuera de la red mallada utilizan la dirección IP que tiene la interfaz del nodo gateway que permite el acceso a Internet. El nodo gateway que une la red mallada inalámbrica con otras redes realiza las funciones de identificación y encaminamiento de los paquetes hacia dentro o fuera de la red mallada inalámbrica según corresponda.

## Conclusiones.

En este Proyecto Fin de Carrera hemos explicado los conceptos básicos de las redes malladas inalámbricas y sus principales ventajas como el bajo coste y la facilidad de despliegue. Estas características son atractivas para la industria y la sociedad. Además existen diversos fabricantes que ofrecen soluciones viables para la implementación de este tipo de redes.

Otro punto a favor es que las redes malladas inalámbricas son compatibles con las redes existentes y por tanto se convierten en una opción muy deseable para ofrecer nuevos servicios a los clientes, ya que las redes malladas inalámbricas soportan servicios de banda ancha.

Tenemos disponibles una serie de protocolos de encaminamiento para las redes malladas inalámbricas con los que poder seguir investigando y mejorando para conseguir un funcionamiento óptimo de la red.

En este proyecto la mayor parte del tiempo se ha dedicado a desplegar la red mallada inalámbrica en el laboratorio. El motivo ha sido que no había demasiada información sobre el protocolo y el software estaba desactualizado. Por tanto hemos tenido que estudiar diversas soluciones y experimentar con ellas hasta conseguir resolver cada contratiempo encontrado en el camino. Utilizando la guía que hemos elaborado en el Apéndice 1 se puede desplegar, instalar y operar con una red mallada inalámbrica de una manera muy fácil y rápida.

Hemos optado por el estudio y experimentación del protocolo B.A.T.M.A.N. que ofrece un rápido encaminamiento de la información al tener en cuenta el mejor vecino de salto para alcanzar a cada destino y no necesita el conocimiento de toda la ruta.

Otra ventaja que nos proporcionan las redes malladas inalámbricas son que pueden implementarse en múltiples dispositivos baratos y que están al alcance de todos.

Para nuestras pruebas en el laboratorio hemos utilizado los routers Linksys WRT54GL. Y solo hemos tenido que cambiar el firmware para poder operar con ellos con la libertad requerida para instalar los paquetes necesarios.

Una vez que ya instalamos el protocolo B.A.T.M.A.N. y configuramos las interfaces inalámbricas con el mismo modo, ssid y canal, ya tenemos preparada la red mallada inalámbrica para realizar las pruebas oportunas.

Hemos estudiado la robustez de la red ante cambios en la topología, como caídas de enlaces y de nodos. Hemos medido el tiempo que tarda la red en recuperarse ante diferentes situaciones y el resultado ha demostrado que el retardo es mayor en el caso de tener que realizar modificaciones en las tablas de encaminamiento, que en el caso de incluir nuevos datos.

Estos resultados son lógicos ya que es más rápido la detección de un nuevo nodo porque en el momento que se incluye se informa a la red de su existencia a través de mensajes

OGMs. Y en cambio la detección de un nodo caído conlleva mayor retardo ya que se debe detectar la caída y luego recalcular las rutas.

Los futuros proyectos podrían ir encaminados en esta misma línea pero implementando otros tipos de protocolos de encaminamiento y así podríamos estudiar las diferencias con este protocolo. Y generalizar a redes mayores para comparar los resultados teóricos.



## **ANEXO 1: PRESUPUESTO**

<b>1.- Autor:</b>						
MARIA BENITO TRUJILLO						
<b>2.- Departamento:</b>						
TELEMATICA						
<b>3.- Descripción del Proyecto:</b>						
- Título	EVALUACION EXPERIMENTAL DE REDES MALLADAS BASADAS EN EL PROTOCOLO B.A.T.M.A.N.					
- Duración (meses)	16					
Tasa de costes Indirectos:	20%					
<b>4.- Presupuesto total del Proyecto (valores en Euros):</b>						
52.068 Euros						
<b>5.- Desglose presupuestario (costes directos)</b>						
<b>PERSONAL</b>						
Apellidos y nombre	N.I.F. (no rellenar - solo a título informativo)	Categoría	Dedicación (meses) <sup>a)</sup>	(hombres)	Coste hombre mes	Coste (Euro)
BENITO TRUJILLO, MARIA	53454317	Ingeniero	16		2.694,39	43.110,24
					<b>Total</b>	43.110,24
<sup>a)</sup> 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas) Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)						
<b>EQUIPOS</b>						
Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable <sup>d)</sup>	
9 ROUTERS LINKSYS WRT54GL	431,91	100	16	60	115,18	
ETGEAR FS108 SWITCH PROFAS	30,04	100	16	60	8,01	
SWITCH 5 PUERTOS SMC 10/100	9,90	100	16	60	2,64	
PIXMANIA CABLE ETHERNET RJ45	3,00	100	16	60	0,80	
3 REGLETAS 5 ENCHUFES	14,97	100	16	60	3,99	
D-LINK TARJETA PCI ETHERNET	10,90	100	16	60	2,91	
ORDENADOR COMPLETO	549,00	100	16	60	146,40	
					<b>Total</b>	279,93
<sup>d)</sup> Fórmula de cálculo de la Amortización:						
$\frac{A}{B} \times C \times D$	A = nº de meses desde la fecha de facturación en que el equipo es utilizado					
	B = periodo de depreciación (60 meses)					
	C = coste del equipo (sin IVA)					
	D = % del uso que se dedica al proyecto (habitualmente 100%)					
<b>6.- Resumen de costes</b>						
Presupuesto Costes Totales	<b>Presupuesto Costes Totales</b>					
Personal	43.110					
Amortización	280					
Costes Indirectos	8.678					
Total	<b>52.068</b>					

## **APÉNDICE 1: CONFIGURACIÓN DE LOS ROUTERS**

En este Apéndice vamos a incluir un manual explicando paso a paso la configuración de los routers Linksys WRT54GL cuando se quiere desplegar una red mallada inalámbrica.  
[11]

Hemos dividido el manual en 11 pasos:

- PASO 1: Configuración de la tarjeta eth1.
- PASO 2: Conexión del router.
- PASO 3: Acceso al router.
- PASO 4: Cambio de password.
- PASO 5: Cambio de firmware.
- PASO 6: Recuperar el router.
- PASO 7: Cambio de IP.
- PASO 8: Acceso al router sin petición de password.
- PASO 9: Instalación de B.A.T.M.A.N. en los routers
- PASO 10: Configuración de los parámetros de B.A.T.M.A.N.
- PASO 11: Instalación de paquetes adicionales.

### **PASO 1: Configuración de la tarjeta eth1**

La dirección IP de la tarjeta eth1 debe estar contenida en el rango de los routers que vamos a utilizar para poder acceder a ellos. Como los routers vienen por defecto con la dirección 192.168.1.1/24, por este motivo elegimos una IP como 192.168.1.100.

Abrimos en el PC un terminal y configuramos la tarjeta eth1 con el siguiente comando:

```
sudo ifconfig eth1 up
```

Asignamos la dirección IP y la máscara a la tarjeta eth1 del siguiente modo:

```
sudo ifconfig eth1 192.168.1.100 netmask 255.255.255.0
```

### **PASO 2: Conexión del router**

Enchufamos el router a la fuente de alimentación y conectamos el cable Ethernet desde la tarjeta eth1 a cualquiera de los 4 puertos disponibles en el router (y nunca en el puerto de Internet).

### **PASO 3: Acceso al router**

Tenemos 2 opciones para acceder al router, una desde el terminal con el siguiente comando:

```
telnet 192.168.1.1
```

```
login:
```

```
password: admin
```

Y otra desde una página web introduciendo en la URL la dirección 192.168.1.1. y con el mismo login y password.

## PASO 4: Cambio de password

Una vez que accedemos al router cambiamos el login y el password:

- Si accedemos por comandos debemos ejecutar `passw` en el router.
- Si accedemos por web: Pestaña Administration → Management.

The screenshot shows the Linksys WRT54GL router's web interface. The top navigation bar includes 'Administration', 'Setup', 'Wireless', 'Security', 'Access Restrictions', 'Applications & Gaming', and 'Status'. The 'Administration' tab is selected, and the 'Management' sub-tab is active. The 'Router Password' section is highlighted in the left sidebar. The main content area contains the following fields:

- Router Password:** A text input field with a password mask (dots).
- Re-enter to confirm:** A second text input field with a password mask (dots).
- Access Server:** Radio buttons for ☒ HTTP and ☐ HTTPS.
- Wireless Access Web:** Radio buttons for ☒ Enable and ☐ Disable.
- Remote Management:** Radio buttons for ☐ Enable and ☒ Disable.
- Management Port:** A text input field containing '8080'.
- Use https:** A checkbox that is currently unchecked.
- UPnP:** Radio buttons for ☒ Enable and ☐ Disable.

On the right side, there is a blue sidebar with instructions:

- Local Router Access:** You can change the Router's password from here. Enter a new Router password and then type it again in the Re-enter to confirm field to confirm.
- Web Access:** Allows you to configure access options to the router's web utility. [More...](#)
- Remote Router Access:** Allows you to access your router remotely. Choose the port you would like to use. You must change the password to the router if it is still using its default password.
- UPnP:** Used by certain programs to automatically open ports for communication. [More...](#)

Figura A1.1: Cambio de password

Usaremos como login **root** y como password la que queramos.

## PASO 5: Cambio de firmware

Una vez que hemos accedido al router desde una página web elegimos la pestaña de *Administración* y seleccionamos el firmware al que queramos cambiar. [20][21][22][23] Para nuestro proyecto seleccionamos el [openwrt-brcm-2.4-squashfs.trx](http://downloads.openwrt.org/kamikaze/8.09.2/brcm-2.4/openwrt-brcm-2.4-squashfs.trx) que se encuentra en enlace:

<http://downloads.openwrt.org/kamikaze/8.09.2/brcm-2.4/openwrt-brcm-2.4-squashfs.trx>

Pero este tipo de imagen .trx no se puede cambiar en el firmware original, por lo que instalamos un firmware .bin: **Administration → Firmware Upgrade**

<http://downloads.openwrt.org/kamikaze/8.09.2-RC2/brcm47xx/openwrt-wrt54g-squashfs.bin>

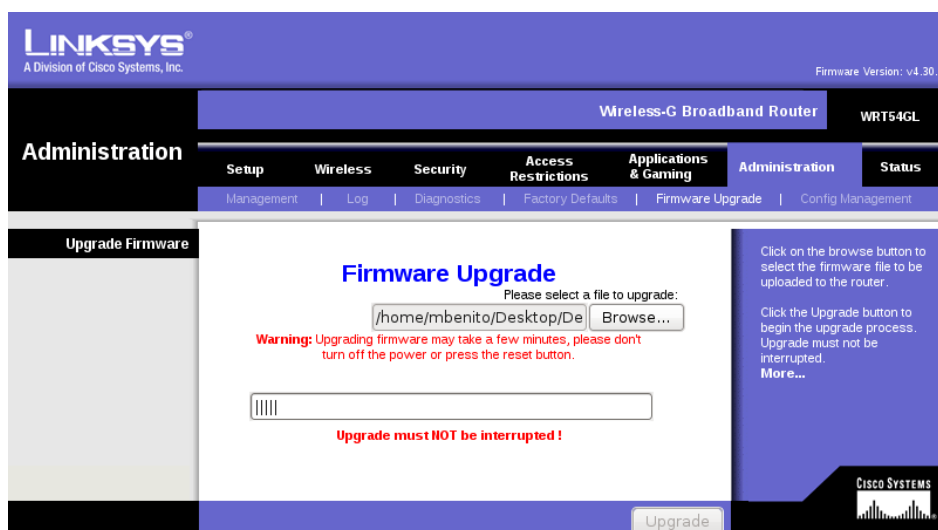


Figura A1.2: Cambio de firmware [openwrt-wrt54g-squashfs.bin](#)

Una vez instalado el firmware [openwrt-wrt54g-squashfs.bin](#) hacemos el upgrade del [openwrt-brcm-2.4-squashfs.trx](#): **System → Flash Firmware**

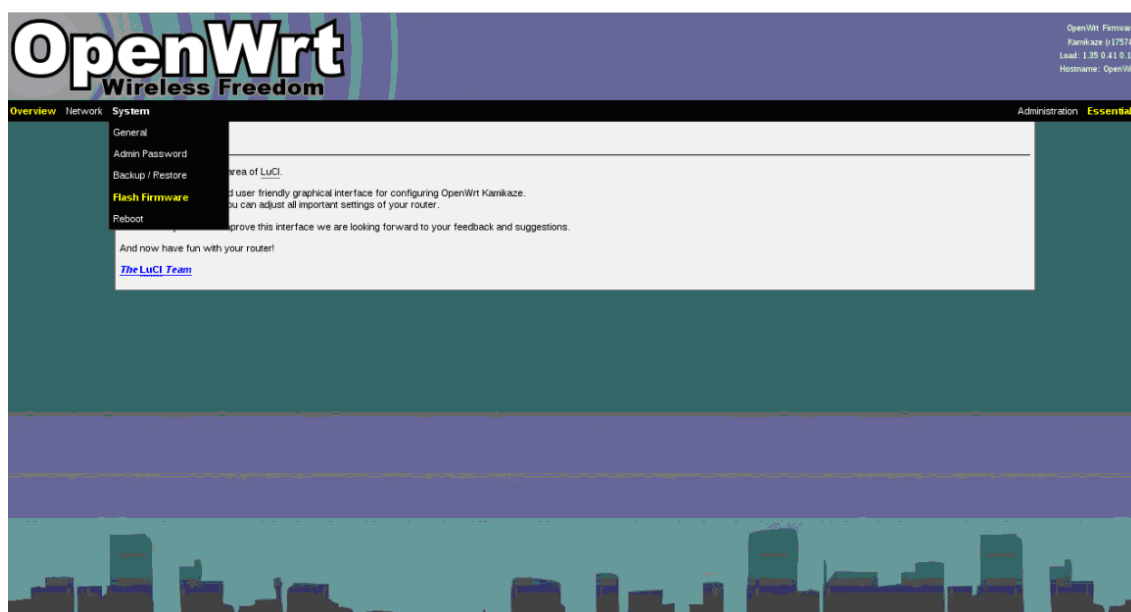


Figura A1.3: Cambio de firmware [openwrt-brcm-2.4-squashfs.trx](#)

Es muy importante que durante el cambio de firmware no se desconecte el router ya que provocaríamos un fallo de carga y el equipo dejaría de responder. Si por cualquier motivo falla la actualización del firmware y el router deja de ser accesible podemos recuperarlo tal y como se explica en el PASO 6.

Ahora cada vez que queramos acceder al router a través de un terminal lo debemos hacer mediante el comando ssh ya que al cambiar de firmware la función telnet se desactiva.

## PASO 6: Recuperar el router

En el caso que la actualización del firmware no haya terminado con éxito y el router “no funcione”, siempre se puede recuperar si podemos llegar al router mediante ping.

Lo primero que debemos hacer es un reset al equipo, esto se consigue pulsando durante unos 30 segundos con un bolígrafo o similar el botón que se encuentra por detrás del router a mano izquierda y que indica Reset.

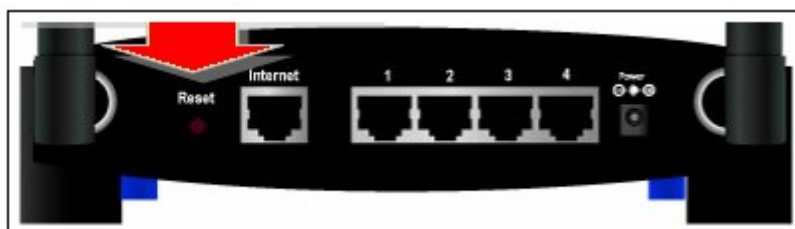


Figura A1.4: Reset router Linksys WRT54GL

Ahora comprobamos que podemos acceder al router mediante ping 192.168.1.1, esta dirección IP es la que tienen todos los routers por defecto.

Procedemos a transferir el firmware utilizando el protocolo de transferencia de ficheros tftp. El firmware ha sido denominado code.bin para que sea más sencillo.

Nos situamos en la carpeta que contenga el code.bin desde un terminal y ahora se ejecutan los siguientes comandos:

```
tftp 192.168.1.1  
binary  
rexmt 1  
timeout 60  
trace on
```

Escribimos put code.bin pero no lo ejecutamos hasta que no hayamos pulsado el botón reset, y manteniéndolo pulsado desconectamos y conectamos la alimentación. Entonces ahora si ejecutamos:

```
put code.bin
```

Una vez que empieza a transmitir el fichero podemos soltar el botón reset. Y una vez que se haya terminado la transferencia podemos acceder al router.

## PASO 7: Acceso al router sin petición de password

Como utilizaremos scripts para ejecutar todos los comandos en los routers de la red, vamos a hacer que el acceso al router mediante ssh no pida contraseña.

Lo que haremos es eliminar la petición de contraseña y añadir la necesidad de identificarse de manera automática mediante una firma RSA que tendremos almacenada.

#Generamos las claves RSA:

```
ssh-keygen -t rsa
```

A continuación nos pregunta la passphrase que en nuestro caso lo dejaremos en blanco.

#Ahora mandamos la clave pública a cada router:

```
scp ~/.ssh/id_dsa.pub root@192.168.1.X:/tmp
```

#Añadimos la clave pública en authorized keys:

```
ssh root@192.168.1.X
```

```
cd /etc/dropbear
```

```
cat /tmp/id_*.pub >> authorized_keys
```

```
chmod 0600 authorized_keys
```

```
exit
```

## PASO 8: Cambio de IP

Disponemos de 9 routers LINKSYS y vamos a asignarle una IP a cada uno de ellos. Para ello nos conectamos al router (mediante una página web e introduciendo 192.168.1.1) y seleccionamos la pestaña *Network*. Asignamos en el campo IPv4-Address la IP que queramos. Utilizaremos las siguientes IPs:

192.168.1.1  
192.168.1.2  
192.168.1.3  
192.168.1.4  
192.168.1.5  
192.168.1.6  
192.168.1.8  
192.168.1.9  
192.168.1.10

The screenshot shows the OpenWrt web interface. The top navigation bar includes 'Overview', 'Network' (highlighted), and 'System'. The main content area is titled 'Network'. Under the 'Status' section, there is a table showing network interfaces:

Network	MAC-Address Hardware Address	IPv4-Address	IPv4-Netmask	Traffic transmitted / received	Errors TX / RX
wan	00:1c:10:a4:2c:57			15.07 KB / 0.00 B	0 / 0
lan	00:1c:10:a4:2c:57	192.168.1.1	255.255.255.0	225.31 KB / 22.96 KB	0 / 0

Below the status table, the 'Local Network' section contains input fields for 'IPv4-Address' (set to 10.10.1.105), 'IPv4-Netmask' (set to 255.255.255.0), 'IPv4-Gateway (optional)', and 'DNS-Server (optional)'. The 'Internet Connection' section shows the 'Protocol' set to 'automatic' and a checked box for 'Fixes problems with unreachable websites, submitting forms or other unexpected behaviour for some ISPs.' At the bottom right, there are 'Reset' and 'Save' buttons.

Figura A1.5: Cambio de IP



## PASO 9: Instalación de B.A.T.M.A.N. en los routers

Uno de los paquetes que están disponibles para el firmware OpenWRT es el demonio de B.A.T.M.A.N. Y lo podemos encontrar en el siguiente enlace:

<http://downloads.openwrt.org/kamikaze/8.09/brcm47xx/packages/>

Nos descargamos el paquete: [batmand\\_r1206-1\\_mipsel.ipk](#).

Transferimos el paquete batmand a cada router:

```
scp batmand_r1206-1_mipsel.ipk root@192.168.1.1:/tmp
scp batmand_r1206-1_mipsel.ipk root@192.168.1.2:/tmp
scp batmand_r1206-1_mipsel.ipk root@192.168.1.3:/tmp
scp batmand_r1206-1_mipsel.ipk root@192.168.1.4:/tmp
scp batmand_r1206-1_mipsel.ipk root@192.168.1.5:/tmp
scp batmand_r1206-1_mipsel.ipk root@192.168.1.6:/tmp
scp batmand_r1206-1_mipsel.ipk root@192.168.1.7:/tmp
scp batmand_r1206-1_mipsel.ipk root@192.168.1.8:/tmp
scp batmand_r1206-1_mipsel.ipk root@192.168.1.9:/tmp
scp batmand_r1206-1_mipsel.ipk root@192.168.1.10:/tmp
```

Ahora nos conectamos a los routers mediante el servidor ssh desde un terminal e instalamos el paquete del protocolo B.A.T.M.A.N.

```
ssh root@192.168.1.1
cd /tmp
opkg install batmand_r1206-1_mipsel.ipk
```

Vemos que para poder instalarlo necesitamos de una librería:

```
root@OpenWrt:/tmp# opkg install batmand_r1206-1_mipsel.ipk
Installing batmand (r1206-1) to root...
Collected errors:
* ERROR: Cannot satisfy the following dependencies for batmand:
    * libpthread * kmod-tun *
root@OpenWrt:/tmp#
```

Por lo que nos descargamos [libpthread\\_0.9.29-14\\_mipsel.ipk](#) que encontramos en <http://downloads.openwrt.org/kamikaze/8.09/brcm47xx/packages/>.

Y necesitamos además el fichero [kmod-tun\\_2.4.35.4-brcm-2.4-1\\_mipsel.ipk](#) que se encuentra en <http://downloads.openwrt.org/kamikaze/8.09.2-RC2/brcm-2.4/packages/>

Transferimos ambos ficheros a los routers con el comando scp.

Instalamos los 3 paquetes en cada router:

```
opkg install kmod-tun_2.4.35.4-brcm-2.4-1_mipsel.ip
opkg install libpthread_0.9.29-14_mipsel.ipk
opkg install batmand_r1206-1_mipsel.ipk
```

## PASO 10: Configuración de los parámetros de B.A.T.M.A.N.

Lo primero que hacemos es visualizar las interfaces wireless disponibles en el router. Esto lo hacemos con el comando *iwconfig*:

```
root@OpenWrt:~# iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

eth0.0  no wireless extensions.

eth0.1  no wireless extensions.

br-lan  no wireless extensions.

wl0     IEEE 802.11-DS  ESSID:"batman-test"
        Mode:Ad-Hoc  Frequency:2.412 GHz  Cell: 9E:7C:08:34:BE:5E
        Bit Rate=54 Mb/s   Tx-Power=32 dBm
        Retry min limit:7   RTS thr:off   Fragment thr:off
        Power Management:off
        Link Quality=5/5  Signal level=-32 dBm  Noise level=-92 dBm
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0  Missed beacon:0

gate0   no wireless extensions.

root@OpenWrt:~#
```

Como podemos observar tenemos creada una interfaz wireless denominada wl0. Para que esta interfaz funcione hay que definir los siguientes parámetros en cada router:

- ESSID: **batman-test**
- Mode: **Ad hoc**
- Channel: **1**

Esto se puede hacer a través de un terminal con el comando:

`ifconfig wl0 mode ad-hoc essid batman-test channel 1`

O a través de la web: Network

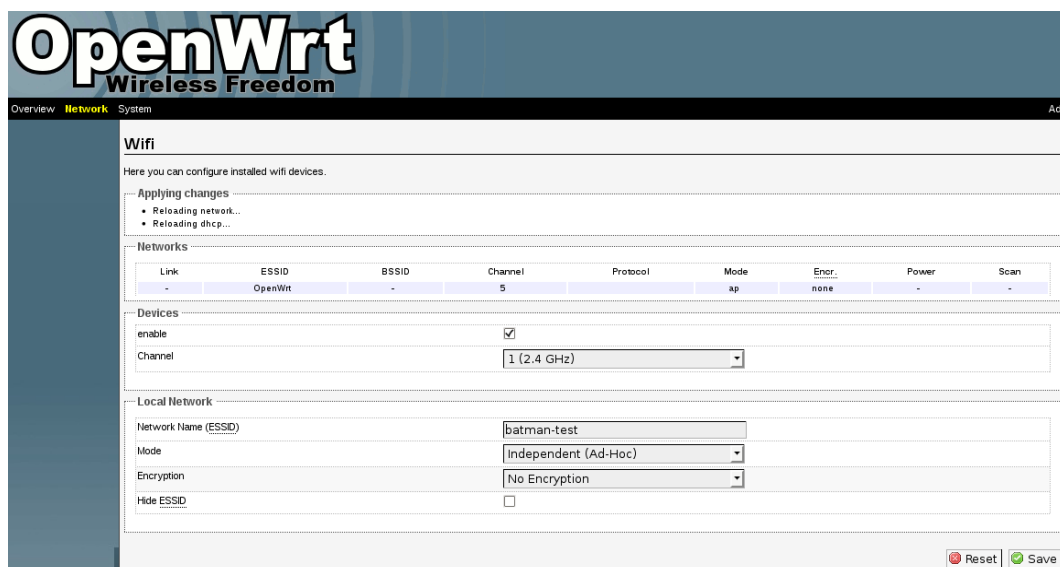


Figura A1.6: Configuración de parámetros

Modificamos los Scripts del paquete batmand para que funcione con nuestros datos:

```
vi /etc/config/wireless: option 'network' 'lan' → option 'network' 'wifi'  
vi /etc/config/batmand: ath0 → wl0
```

Ahora configuramos el alias de la interfaz inalámbrica en cada router y ejecutamos el protocolo B.A.T.M.A.N.:

```
ssh root@192.168.1.X  
ifconfig wl0:bat 10.10.0.X netmask 255.255.0.0 broadcast 10.10.255.255  
batmand -c -d 1 -b
```

Donde X es 1, 2, 3, 4, 5, 6, 7, 8, 9 y 10.

Explicación del comando *batmand -c -d 1 -b*. La opción *-c* llama al batmand para conectarse al demonio batmand que está ejecutándose. La opción *-d 1* es para imprimir la información depurada a primer nivel. La opción *-b* se utiliza para que la información se proporcione una sola vez. Si esta opción no se ejecuta la información de depuración se actualiza cada segundo y se para presionando las teclas Ctrl-C.

```
root@OpenWrt:~# batmand -c -d 1 -b  
Originator (#/255) Nextthop [outgoingIF]: Potential nexthops ... [B.A.T.M.A.N. 0.3.1 rv1206, MainIF/IP: wl0:bat/10.10.0.1, UT: 0d  
0h 0m]  
10.10.0.5 (109) 10.10.0.3 [wl0:bat]: 10.10.0.5 ( 0) 10.10.0.2 ( 96) 10.10.0.3 (109)  
10.10.0.4 (121) 10.10.0.2 [wl0:bat]: 10.10.0.2 (121) 10.10.0.3 ( 51)  
10.10.0.2 (191) 10.10.0.2 [wl0:bat]: 10.10.0.2 (191) 10.10.0.3 ( 81)  
10.10.0.3 (162) 10.10.0.3 [wl0:bat]: 10.10.0.3 (162) 10.10.0.2 ( 78)  
10.10.0.9 ( 46) 10.10.0.2 [wl0:bat]: 10.10.0.2 ( 46) 10.10.0.3 ( 46)  
10.10.0.10 ( 25) 10.10.0.2 [wl0:bat]: 10.10.0.2 ( 25) 10.10.0.3 ( 18)  
10.10.0.6 (110) 10.10.0.3 [wl0:bat]: 10.10.0.3 (110) 10.10.0.2 ( 53)  
10.10.0.8 ( 60) 10.10.0.2 [wl0:bat]: 10.10.0.2 ( 60) 10.10.0.3 ( 42)  
root@OpenWrt:~#
```

La primera línea especifica el tipo de B.A.T.M.A.N. utilizado, la etiqueta y la IP del interfaz utilizado, el tamaño de ventana, el tiempo de espera de enlace bidireccional, el intervalo de inicio y el tiempo que ha pasado desde que se ha iniciado el proceso.

En la siguiente línea muestra las cabeceras de la tabla. Estas líneas comienzan con la IP del nodo B.A.T.M.A.N., a continuación se describe el interfaz usado en el routing para llegar al nodo, y el mejor vecino para llegar al nodo.

## PASO 11: Instalación de paquetes adicionales

Necesitamos instalar otros paquetes para comprobar el funcionamiento del protocolo B.A.T.M.A.N. Los comandos que necesitamos son *ip route*, *tcpdump* y *fping*, y los podemos ejecutar si instalamos *ip\_2.6.25-1\_mipsel.ipk*, *tcpdump\_3.9.8-1.1\_mipsel.ipk* y *fping\_2.4b2\_to-ipv6-1\_mipsel.ipk*. Estos paquetes los encontramos en:

<http://downloads.openwrt.org/kamikaze/8.09/brcm47xx/packages/>

Para instalar tcpdump necesitamos además la librería [libpcap\\_0.9.8-1\\_mipsel.ipk](#) que se encuentra en el mismo directorio anterior.

Como ya hemos explicado antes los pasos para la instalación son:

- Descargamos los paquetes.
- Abrimos un terminal y hacemos la transferencia al router con el comando scp:

```
scp ip_2.6.25-1_mipsel.ipk root@192.168.1.X:./tmp
scp libpcap_0.9.8-1_mipsel.ipk root@192.168.1.X:./tmp
scp tcpdump_3.9.8-1.1_mipsel.ipk root@192.168.1.X:./tmp
scp fping_2.4b2_to-ipv6-1_mipsel.ipk root@192.168.1.X:./tmp
```

- Nos conectamos al router e instalamos los paquetes:

```
ssh root@192.168.1.X
opkg install ip_2.6.25-1_mipsel.ipk
opkg install libpcap_0.9.8-1_mipsel.ipk
opkg install tcpdump_3.9.8-1.1_mipsel.ipk
opkg install fping_2.4b2_to-ipv6-1_mipsel.ipk
```

## **APÉNDICE 2: SCRIPTS Y RESULTADOS**

En este capítulo vamos a incluir todos los Scripts y resultados por comandos que se han obtenido en el proyecto. Disponemos de varios Scripts que nos permiten automatizar las configuraciones de los nodos y comprobar que los resultados son los esperados.

Además incluiremos las capturas de tráfico dirigido a Internet desde dentro y fuera de la red mallada inalámbrica.

## A2.1. SCRIPT CONEXIÓN A INTERNET

El primer Script que vamos a incluir se denomina “Conexion a Internet”. Configura la gateway y los clientes, y por supuesto activa el demonio batmand en cada uno de ellos. Además indica los nodos que no son vecinos para definir la distribución que hemos elegido (Figura 4.3). Utilizamos las direcciones MAC para definir la distribución porque si utilizamos las direcciones IP no se obtendría información de ese nodo, y lo que queremos es que no se reciba información directamente de ese nodo pero si a través de otros nodos. La idea es recibir toda la información de las direcciones MAC de los vecinos independientemente del nodo que ha generado la información. A continuación mostramos el Script “Conexion a Internet”:

### #GATEWAY#

#### echo "GATEWAY ROUTER 1"

```
ssh root@192.168.1.1 '/etc/init.d/firewall stop ; Killall B.A.T.M.A.N.d ; /etc/init.d/B.A.T.M.A.N.d stop ;  
udhcpc -i eth0.1 ; iptables -t nat -I POSTROUTING 1 -o eth0.1 -j MASQUERADE ; iptables -L -t nat -n  
-vv ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:98:4C -j REJECT ; iptables -A INPUT -m  
mac --mac-source 00:1C:10:9A:72:8D -j REJECT ; iptables -A INPUT -m mac --mac-source  
00:1C:10:88:96:06 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:2C:3E -j REJECT  
; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:37:8A -j REJECT ; ifconfig wl0:bat 10.10.0.1  
netmask 255.255.0.0 broadcast 10.10.255.255 ; B.A.T.M.A.N.d -g 5000 wl0:bat'
```

### #CLIENTES#

#### echo "CLIENTE 2 no vecino de 3 6 8 9 10"

```
ssh root@192.168.1.2 '/etc/init.d/firewall stop ; iptables -A INPUT -m mac --mac-source  
00:1C:10:A4:36:67 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:9A:72:8D -j  
REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:96:06 -j REJECT ; iptables -A INPUT  
-m mac --mac-source 00:1C:10:A4:2C:3E -j REJECT ; iptables -A INPUT -m mac --mac-source  
00:1C:10:A4:37:8A -j REJECT ; killall B.A.T.M.A.N.d ; /etc/init.d/B.A.T.M.A.N.d stop ; ifconfig  
wl0:bat 10.10.0.2 netmask 255.255.0.0 broadcast 10.10.255.255 ; B.A.T.M.A.N.d -r 3 wl0:bat '
```

#### echo "CLIENTE 3 no vecino de 2 4 8 9 10"

```
ssh root@192.168.1.3 '/etc/init.d/firewall stop ; iptables -A INPUT -m mac --mac-source  
00:1C:10:88:A3:A7 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:98:4C -j  
REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:96:06 -j REJECT ; iptables -A INPUT  
-m mac --mac-source 00:1C:10:A4:2C:3E -j REJECT ; iptables -A INPUT -m mac --mac-source  
00:1C:10:A4:37:8A -j REJECT ; killall B.A.T.M.A.N.d ; /etc/init.d/B.A.T.M.A.N.d stop ; ifconfig  
wl0:bat 10.10.0.3 netmask 255.255.0.0 broadcast 10.10.255.255 ; B.A.T.M.A.N.d -r 3 wl0:bat '
```

#### echo "CLIENTE 4 no vecino de 1 3 5 6 9 10"

```
ssh root@192.168.1.4 '/etc/init.d/firewall stop ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:36:43 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:36:67 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:2C:59 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:9A:72:8D -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:2C:3E -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:37:8A -j REJECT ; killall B.A.T.M.A.N.d ; /etc/init.d/B.A.T.M.A.N.d stop ; ifconfig wl0:bat 10.10.0.4 netmask 255.255.0.0 broadcast 10.10.255.255 ; B.A.T.M.A.N.d -r 3 wl0:bat '
```

**echo "CLIENTE 5 no vecino de 1 4 6 10"**

```
ssh root@192.168.1.5 '/etc/init.d/firewall stop ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:36:43 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:98:4C -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:9A:72:8D -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:37:8A -j REJECT ; killall B.A.T.M.A.N.d ; /etc/init.d/B.A.T.M.A.N.d stop ; ifconfig wl0:bat 10.10.0.5 netmask 255.255.0.0 broadcast 10.10.255.255 ; B.A.T.M.A.N.d -r 3 wl0:bat '
```

**echo "CLIENTE 6 no vecino de 1 2 4 5 8 10"**

```
ssh root@192.168.1.6 '/etc/init.d/firewall stop ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:36:43 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:A3:A7 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:98:4C -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:2C:59 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:96:06 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:37:8A -j REJECT ; killall B.A.T.M.A.N.d ; /etc/init.d/B.A.T.M.A.N.d stop ; ifconfig wl0:bat 10.10.0.6 netmask 255.255.0.0 broadcast 10.10.255.255 ; B.A.T.M.A.N.d -r 3 wl0:bat '
```

**echo "CLIENTE 8 no vecino de 1 2 3 6 9"**

```
ssh root@192.168.1.8 '/etc/init.d/firewall stop ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:36:43 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:A3:A7 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:36:67 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:9A:72:8D -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:2C:3E -j REJECT ; killall B.A.T.M.A.N.d ; /etc/init.d/B.A.T.M.A.N.d stop ; ifconfig wl0:bat 10.10.0.8 netmask 255.255.0.0 broadcast 10.10.255.255 ; B.A.T.M.A.N.d -r 3 wl0:bat '
```

**echo "CLIENTE 9 no vecino de 1 2 3 4 8"**

```
ssh root@192.168.1.9 '/etc/init.d/firewall stop ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:36:43 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:A3:A7 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:36:67 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:98:4C -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:96:06 -j REJECT ; killall B.A.T.M.A.N.d ; /etc/init.d/B.A.T.M.A.N.d stop ; ifconfig wl0:bat 10.10.0.9 netmask 255.255.0.0 broadcast 10.10.255.255 ; B.A.T.M.A.N.d -r 3 wl0:bat '
```

**#echo "CLIENTE 10 no vecino de 1 2 3 4 5 6"**

```
ssh root@192.168.1.10 '/etc/init.d/firewall stop ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:36:43 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:A3:A7 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:36:67 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:88:98:4C -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:A4:2C:59 -j REJECT ; iptables -A INPUT -m mac --mac-source 00:1C:10:9A:72:8D -j REJECT ; killall B.A.T.M.A.N.d ; /etc/init.d/B.A.T.M.A.N.d stop ; ifconfig wl0:bat 10.10.0.10 netmask 255.255.0.0 broadcast 10.10.255.255 ; B.A.T.M.A.N.d -r 3 wl0:bat '
```

**#Enviamos a cada router en el /etc/resolv.conf un fichero con el mismo nombre y que incluya la IP 163.117.139.253**

```
i=2  
  
for i in 2 3 4 5 6 8 9 10; do  
  
scp resolv.conf root@192.168.1.$i:/etc  
  
done
```

En redes con muchos nodos y cambios frecuentes una opción más fácil sería configurar los nodos vecinos con ACCEPT y REJECT todos los demás. Para nuestro caso con una topología fija nos es similar.

## A2.2. SCRIPT COMPROBACIÓN DE VECINOS

El siguiente Script es denominado “Comprobacion de vecinos” y comprueba consultando las tablas de encaminamiento que los vecinos son los que se describen en la Figura 4.3.

```
echo "router 1 vecino de 2 y 3"  
ssh -l root 192.168.1.1 'ip route list table 66'  
  
echo "router 2 vecino de 1, 4 y 5"  
ssh -l root 192.168.1.2 'ip route list table 66'  
  
echo "router 3 vecino de 1, 5 y 6"  
ssh -l root 192.168.1.3 'ip route list table 66'  
  
echo "router 4 vecino de 2 y 8"  
ssh -l root 192.168.1.4 'ip route list table 66'  
  
echo "router 5 vecino de 2, 3, 8 y 9"  
ssh -l root 192.168.1.5 'ip route list table 66'  
  
echo "router 6 vecino de 3 y 9"  
ssh -l root 192.168.1.6 'ip route list table 66'  
  
echo "router 8 vecino de 4, 5 y 10"  
ssh -l root 192.168.1.8 'ip route list table 66'  
  
echo "router 9 vecino de 5, 6 y 10"  
ssh -l root 192.168.1.9 'ip route list table 66'  
  
echo "router 10 vecino de 8 y 9"  
ssh -l root 192.168.1.10 'ip route list table 66'
```

A continuación presentamos la salida del Script anterior “Comprobacion de vecinos”. Como podemos observar para cada nodo se indica los vecinos que puede alcanzar a través de la interfaz inalámbrica wl0. Y el resto de nodos que se pueden alcanzar mediante uno de los vecinos.

```
mbenito@liendre:~/Desktop$ sh Comprobacion\ de\ vecinos  
router 1 vecino de 2 y 3  
10.10.0.3 dev wl0 proto static scope link src 10.10.0.1
```



10.10.0.2 dev wl0 proto static scope link src 10.10.0.1  
10.10.0.5 via 10.10.0.3 dev wl0 proto static src 10.10.0.1  
10.10.0.4 via 10.10.0.2 dev wl0 proto static src 10.10.0.1  
10.10.0.6 via 10.10.0.3 dev wl0 proto static src 10.10.0.1  
10.10.0.9 via 10.10.0.3 dev wl0 proto static src 10.10.0.1  
10.10.0.8 via 10.10.0.3 dev wl0 proto static src 10.10.0.1  
10.10.0.10 via 10.10.0.3 dev wl0 proto static src 10.10.0.1

**router 2 vecino de 1, 4 y 5**

10.10.0.1 dev wl0 proto static scope link src 10.10.0.2  
10.10.0.3 via 10.10.0.1 dev wl0 proto static src 10.10.0.2  
10.10.0.5 dev wl0 proto static scope link src 10.10.0.2  
10.10.0.4 dev wl0 proto static scope link src 10.10.0.2  
10.10.0.6 via 10.10.0.1 dev wl0 proto static src 10.10.0.2  
10.10.0.9 via 10.10.0.5 dev wl0 proto static src 10.10.0.2  
10.10.0.8 via 10.10.0.5 dev wl0 proto static src 10.10.0.2  
10.10.0.10 via 10.10.0.5 dev wl0 proto static src 10.10.0.2

**router 3 vecino de 1, 5 y 6**

10.10.0.1 dev wl0 proto static scope link src 10.10.0.3  
10.10.0.2 via 10.10.0.1 dev wl0 proto static src 10.10.0.3  
10.10.0.5 dev wl0 proto static scope link src 10.10.0.3  
10.10.0.4 via 10.10.0.5 dev wl0 proto static src 10.10.0.3  
10.10.0.6 dev wl0 proto static scope link src 10.10.0.3  
10.10.0.9 via 10.10.0.5 dev wl0 proto static src 10.10.0.3  
10.10.0.8 via 10.10.0.5 dev wl0 proto static src 10.10.0.3  
10.10.0.10 via 10.10.0.5 dev wl0 proto static src 10.10.0.3

**router 4 vecino de 2 y 8**

10.10.0.1 via 10.10.0.2 dev wl0 proto static src 10.10.0.4  
10.10.0.3 via 10.10.0.2 dev wl0 proto static src 10.10.0.4  
10.10.0.2 dev wl0 proto static scope link src 10.10.0.4  
10.10.0.5 via 10.10.0.2 dev wl0 proto static src 10.10.0.4  
10.10.0.6 via 10.10.0.2 dev wl0 proto static src 10.10.0.4  
10.10.0.9 via 10.10.0.2 dev wl0 proto static src 10.10.0.4  
10.10.0.8 dev wl0 proto static scope link src 10.10.0.4  
10.10.0.10 via 10.10.0.8 dev wl0 proto static src 10.10.0.4

**router 5 vecino de 2, 3, 8 y 9**

10.10.0.1 via 10.10.0.3 dev wl0 proto static src 10.10.0.5  
10.10.0.3 dev wl0 proto static scope link src 10.10.0.5  
10.10.0.2 dev wl0 proto static scope link src 10.10.0.5  
10.10.0.4 via 10.10.0.8 dev wl0 proto static src 10.10.0.5  
10.10.0.6 via 10.10.0.3 dev wl0 proto static src 10.10.0.5  
10.10.0.9 dev wl0 proto static scope link src 10.10.0.5  
10.10.0.8 dev wl0 proto static scope link src 10.10.0.5  
10.10.0.10 via 10.10.0.9 dev wl0 proto static src 10.10.0.5

**router 6 vecino de 3 y 9**

10.10.0.1 via 10.10.0.3 dev wl0 proto static src 10.10.0.6  
10.10.0.3 dev wl0 proto static scope link src 10.10.0.6  
10.10.0.2 via 10.10.0.3 dev wl0 proto static src 10.10.0.6  
10.10.0.5 via 10.10.0.9 dev wl0 proto static src 10.10.0.6  
10.10.0.4 via 10.10.0.3 dev wl0 proto static src 10.10.0.6  
10.10.0.9 dev wl0 proto static scope link src 10.10.0.6  
10.10.0.8 via 10.10.0.3 dev wl0 proto static src 10.10.0.6  
10.10.0.10 via 10.10.0.9 dev wl0 proto static src 10.10.0.6

**router 8 vecino de 4, 5 y 10**

10.10.0.1 via 10.10.0.4 dev wl0 proto static src 10.10.0.8  
10.10.0.3 via 10.10.0.5 dev wl0 proto static src 10.10.0.8  
10.10.0.2 via 10.10.0.4 dev wl0 proto static src 10.10.0.8  
10.10.0.5 dev wl0 proto static scope link src 10.10.0.8  
10.10.0.4 dev wl0 proto static scope link src 10.10.0.8  
10.10.0.6 via 10.10.0.10 dev wl0 proto static src 10.10.0.8  
10.10.0.9 via 10.10.0.10 dev wl0 proto static src 10.10.0.8

```

10.10.0.10 dev wl0 proto static scope link src 10.10.0.8
router 9 vecino de 5, 6 y 10
10.10.0.1 via 10.10.0.5 dev wl0 proto static src 10.10.0.9
10.10.0.3 via 10.10.0.5 dev wl0 proto static src 10.10.0.9
10.10.0.2 via 10.10.0.5 dev wl0 proto static src 10.10.0.9
10.10.0.5 dev wl0 proto static scope link src 10.10.0.9
10.10.0.4 via 10.10.0.5 dev wl0 proto static src 10.10.0.9
10.10.0.6 dev wl0 proto static scope link src 10.10.0.9
10.10.0.8 via 10.10.0.5 dev wl0 proto static src 10.10.0.9
10.10.0.10 dev wl0 proto static scope link src 10.10.0.9
router 10 vecino de 8 y 9
10.10.0.1 via 10.10.0.9 dev wl0 proto static src 10.10.0.10
10.10.0.3 via 10.10.0.9 dev wl0 proto static src 10.10.0.10
10.10.0.2 via 10.10.0.9 dev wl0 proto static src 10.10.0.10
10.10.0.5 via 10.10.0.9 dev wl0 proto static src 10.10.0.10
10.10.0.4 via 10.10.0.8 dev wl0 proto static src 10.10.0.10
10.10.0.6 via 10.10.0.9 dev wl0 proto static src 10.10.0.10
10.10.0.9 dev wl0 proto static scope link src 10.10.0.10
10.10.0.8 dev wl0 proto static scope link src 10.10.0.10

```

## A2.3. COMPROBAR EL FUNCIONAMIENTO DE LA RED

Una prueba muy sencilla y rápida para comprobar que la red funciona una vez que se ha configurado y se han comprobado los vecinos es realizar un ping desde uno de los nodos al resto de nodos de la red mallada inalámbrica. Hemos elegido el router 192.168.1.1 para realizar un ping a todas las interfaces inalámbricas de B.A.T.M.A.N.:

```
mbenito@liendre:~/Desktop$ ssh root@192.168.1.1
```

```

BusyBox v1.11.2 (2009-01-06 07:18:07 CET) built-in shell (ash)
Enter 'help' for a list of built-in commands.

```

```

  _____
 | | | | | | | | | | | | | | | | | | | | | |
 | - || _ | - || || || _|| _|
 |_____| |_____| |_____| |_____| |_____|
 | | W I R E L E S S F R E E D O M
 KAMIKAZE (8.09, r14511) -----
 * 10 oz Vodka      Shake well with ice and strain
 * 10 oz Triple sec mixture into 10 shot glasses.
 * 10 oz lime juice Salute!

```

```

root@OpenWrt:~# ping 10.10.0.2
PING 10.10.0.2 (10.10.0.2): 56 data bytes
64 bytes from 10.10.0.2: seq=0 ttl=64 time=5.091 ms

```

```

--- 10.10.0.2 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 5.091/5.091/5.091 ms

```

```

root@OpenWrt:~# ping 10.10.0.3
PING 10.10.0.3 (10.10.0.3): 56 data bytes
64 bytes from 10.10.0.3: seq=0 ttl=64 time=3.128 ms

```

```

--- 10.10.0.3 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss

```

```
round-trip min/avg/max = 3.128/3.128/3.128 ms
root@OpenWrt:~# ping 10.10.0.4
PING 10.10.0.4 (10.10.0.4): 56 data bytes
64 bytes from 10.10.0.4: seq=0 ttl=63 time=3.643 ms

--- 10.10.0.4 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 3.643/3.643/3.643 ms
root@OpenWrt:~# ping 10.10.0.5
PING 10.10.0.5 (10.10.0.5): 56 data bytes
64 bytes from 10.10.0.5: seq=0 ttl=63 time=8.088 ms

--- 10.10.0.5 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 8.088/8.088/8.088 ms
root@OpenWrt:~# ping 10.10.0.6
PING 10.10.0.6 (10.10.0.6): 56 data bytes
64 bytes from 10.10.0.6: seq=0 ttl=63 time=5.130 ms
64 bytes from 10.10.0.6: seq=1 ttl=63 time=12.215 ms

--- 10.10.0.6 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 5.130/8.672/12.215 ms
root@OpenWrt:~# ping 10.10.0.8
PING 10.10.0.8 (10.10.0.8): 56 data bytes
64 bytes from 10.10.0.8: seq=0 ttl=62 time=11.192 ms

--- 10.10.0.8 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 11.192/11.192/11.192 ms
root@OpenWrt:~# ping 10.10.0.9
PING 10.10.0.9 (10.10.0.9): 56 data bytes
64 bytes from 10.10.0.9: seq=0 ttl=62 time=19.992 ms

--- 10.10.0.9 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 19.992/19.992/19.992 ms
root@OpenWrt:~# ping 10.10.0.10
PING 10.10.0.10 (10.10.0.10): 56 data bytes
64 bytes from 10.10.0.10: seq=0 ttl=61 time=20.352 ms
64 bytes from 10.10.0.10: seq=1 ttl=61 time=6.032 ms

--- 10.10.0.10 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 6.032/13.192/20.352 ms
root@OpenWrt:~#
```

## A2.4. COMPROBAR EL ACCESO A INTERNET

Para comprobar que todos los nodos de la red mallada inalámbrica tienen acceso a Internet vamos a realizar un ping a una página web desde todos los nodos.

```
mbenito@liendre:~/Desktop$ ssh root@192.168.1.1
```

```
BusyBox v1.11.2 (2009-01-06 07:18:07 CET) built-in shell (ash)
Enter 'help' for a list of built-in commands.
```

KAMIKAZE (8.09, r14511) -----

- \* 10 oz Vodka      Shake well with ice and strain
- \* 10 oz Triple sec    mixture into 10 shot glasses.
- \* 10 oz lime juice    Salute!

```
root@OpenWrt:~# ping google.es
```

PING google.es (74.125.77.104): 56 data bytes  
64 bytes from 74.125.77.104: seq=0 ttl=49 time=42.391 ms

--- google.es ping statistics ---

1 packets transmitted, 1 packets received, 0% packet loss  
round-trip min/avg/max = 42.391/42.391/42.391 ms

```
root@OpenWrt:~# exit
```

Connection to 192.168.1.1 closed.

```
mbenito@liendre:~/Desktop$ ssh root@192.168.1.2
```

```
BusyBox v1.11.2 (2009-01-06 07:18:07 CET) built-in shell (ash)
Enter 'help' for a list of built-in commands.
```



KAMIKAZE (8.09, r14511) -----

- \* 10 oz Vodka      Shake well with ice and strain
- \* 10 oz Triple sec    mixture into 10 shot glasses.
- \* 10 oz lime juice    Salute!

```
root@OpenWrt:~# ping google.es
```

PING google.es (74.125.77.104): 56 data bytes  
64 bytes from 74.125.77.104: seq=0 ttl=48 time=48.982 ms

--- google.es ping statistics ---

1 packets transmitted, 1 packets received, 0% packet loss  
round-trip min/avg/max = 48.982/48.982/48.982 ms

```
root@OpenWrt:~# exit
```

Connection to 192.168.1.2 closed.

```
mbenito@liendre:~/Desktop$ ssh root@192.168.1.3
```

BusyBox v1.11.2 (2009-01-06 07:18:07 CET) built-in shell (ash)

- \* 10 oz Vodka      Shake well with ice and strain
- \* 10 oz Triple sec   mixture into 10 shot glasses.
- \* 10 oz lime juice   Salute!

Connection to 192.168.1.3 closed.

WIRELESS FREEDOM

- \* 10 oz Vodka      Shake well with ice and strain
- \* 10 oz Triple sec   mixture into 10 shot glasses.
- \* 10 oz lime juice   Salute!

Connection to 192.168.1.4 closed.

WIRELESS FREEDOM

- \* 10 oz Vodka      Shake well with ice and strain
- \* 10 oz Triple sec    mixture into 10 shot glasses.
- \* 10 oz lime juice    Salute!

```
-- google.es ping statistics --
2 packets transmitted, 1 packets received, 50% packet loss
round-trip min/avg/max = 74.354/74.354/74.354 ms
root@OpenWrt:~# exit
Connection to 192.168.1.5 closed.
```

```
BusyBox v1.11.2 (2009-01-06 07:18:07 CET) built-in shell (ash)
Enter 'help' for a list of built-in commands.
```

```
root@OpenWrt:~# ping google.es
PING google.es (216.239.59.104): 56 data bytes
64 bytes from 216.239.59.104: seq=0 ttl=49 time=61.180 ms
```

```

--- google.es ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 61.180/61.180/61.180 ms
root@OpenWrt:~# exit
Connection to 192.168.1.6 closed.

```

```
BusyBox v1.11.2 (2009-01-06 07:18:07 CET) built-in shell (ash)
Enter 'help' for a list of built-in commands.
```

```
root@OpenWrt:~# ping google.es
PING google.es (209.85.229.104): 56 data bytes
64 bytes from 209.85.229.104: seq=0 ttl=50 time=33.761 ms
```

```

--- google.es ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 33.761/33.761/33.761 ms
root@OpenWrt:~# exit
Connection to 192.168.1.8 closed.

```

BusyBox v1.11.2 (2009-01-06 07:18:07 CET) built-in shell (ash)

```
|_|.|-----|.|||.|.|_|_| | | |
|-||_|-_||||_|_|_|_|_|  
|_|_|_|_|_|_|_|_|_|_|_|_|_|  
|_| WIRELESS FREEDOM  
KAMIKAZE (8.09, r14511) -----  
* 10 oz Vodka      Shake well with ice and strain  
* 10 oz Triple sec mixture into 10 shot glasses.  
* 10 oz lime juice Salute!  
  
-----  
root@OpenWrt:~# ping google.es  
PING google.es (74.125.77.104): 56 data bytes  
64 bytes from 74.125.77.104: seq=0 ttl=48 time=50.814 ms  
  
--- google.es ping statistics ---  
1 packets transmitted, 1 packets received, 0% packet loss  
round-trip min/avg/max = 50.814/50.814/50.814 ms  
root@OpenWrt:~# exit  
Connection to 192.168.1.9 closed.
```

```
BusyBox v1.11.2 (2009-01-06 07:18:07 CET) built-in shell (ash)
Enter 'help' for a list of built-in commands.
```

[illegible]

## A2.5. MEDIDAS DEL RETARDO CUANDO ARRANCA LA RED

Para obtener estas medidas hemos creado el Script “arranque” que básicamente consiste en conectarse al nodo y ejecutar un bucle con el comando fping y date. Los datos que conseguimos de estos comandos es saber la hora exacta que se ha ejecutado el Script y el momento en que el comando fping ha conseguido acceder al nodo.

```
ssh root@192.168.1.1 'a=1;y=500;while [ $a -ne $y ];do date;fping 10.10.0.10 -i 1 -c 10;a=`expr $a + 1`;done '
```

Los resultados de las 30 medidas en la situación de arranque de la red son las siguientes:

```
mбенито@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:51:21 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:51:22 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:51:23 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:51:24 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:51:25 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
mбенито@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:51:26 UTC 2009
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [0], 84 bytes, 3045 ms (3045 avg, 75% loss)
10.10.0.10 : [1], 84 bytes, 2036 ms (2541 avg, 50% loss)
10.10.0.10 : [2], 84 bytes, 1028 ms (2036 avg, 25% loss)
10.10.0.10 : [3], 84 bytes, 21.3 ms (1532 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 12.2 ms (1228 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 18.5 ms (1027 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 1009 ms (1024 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 5.54 ms (897 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 19.8 ms (799 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 24.4 ms (722 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 5.54/722/3045
Fri Feb 6 02:51:36 UTC 2009
^Cmбенито@liendre:~/Desktop$
```

### RETARDO = 5 segundos

```
mбенито@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:50:18 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:19 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
mбенито@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:50:20 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:21 UTC 2009
```



10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:22 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
mbenito@liendre:~/Desktop\$ sh arranque  
Fri Feb 6 02:50:23 UTC 2009

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
10.10.0.10 : [0], 84 bytes, 1177 ms (1177 avg, 50% loss)  
10.10.0.10 : [1], 84 bytes, 171 ms (674 avg, 0% loss)  
10.10.0.10 : [2], 84 bytes, 11.2 ms (453 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 92.7 ms (363 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 1030 ms (496 avg, 16% loss)  
10.10.0.10 : [5], 84 bytes, 21.6 ms (417 avg, 0% loss)  
10.10.0.10 : [6], 84 bytes, 12.6 ms (359 avg, 0% loss)  
10.10.0.10 : [7], 84 bytes, 23.2 ms (317 avg, 0% loss)  
10.10.0.10 : [8], 84 bytes, 46.8 ms (287 avg, 0% loss)  
10.10.0.10 : [9], 84 bytes, 195 ms (278 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 11.2/278/1177  
Fri Feb 6 02:50:33 UTC 2009  
^Cmbenito@liendre:~/Desktop\$

## RETARDO = 5 segundos

mbenito@liendre:~/Desktop\$ sh arranque  
Fri Feb 6 02:50:23 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:24 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:25 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:26 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:27 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:28 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:29 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:30 UTC 2009  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
10.10.0.10 : [0], 84 bytes, 1008 ms (1008 avg, 50% loss)  
10.10.0.10 : [1], 84 bytes, 6.75 ms (507 avg, 0% loss)  
10.10.0.10 : [2], 84 bytes, 15.2 ms (343 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 26.6 ms (264 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 31.7 ms (217 avg, 0% loss)  
10.10.0.10 : [5], 84 bytes, 11.1 ms (183 avg, 0% loss)  
10.10.0.10 : [6], 84 bytes, 6.78 ms (158 avg, 0% loss)  
10.10.0.10 : [7], 84 bytes, 14.7 ms (140 avg, 0% loss)  
10.10.0.10 : [8], 84 bytes, 39.3 ms (129 avg, 0% loss)  
10.10.0.10 : [9], 84 bytes, 21.4 ms (118 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 6.75/118/1008  
Fri Feb 6 02:50:39 UTC 2009  
^Cmbenito@liendre:~/Desktop\$

## RETARDO = 7 segundos

mбенито@liendre:~/Desktop\$ sh arranque  
Fri Feb 6 02:51:00 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:51:01 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:51:02 UTC 2009  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
10.10.0.10 : [3], 84 bytes, 1037 ms (1037 avg, 80% loss)  
10.10.0.10 : [4], 84 bytes, 38.6 ms (538 avg, 60% loss)  
10.10.0.10 : [5], 84 bytes, 7.70 ms (361 avg, 50% loss)  
10.10.0.10 : [6], 84 bytes, 71.3 ms (288 avg, 42% loss)  
10.10.0.10 : [7], 84 bytes, 13.3 ms (233 avg, 37% loss)  
10.10.0.10 : [8], 84 bytes, 6.58 ms (195 avg, 33% loss)  
10.10.0.10 : [9], 84 bytes, 11.0 ms (169 avg, 30% loss)

10.10.0.10 : xmt/rcv/%loss = 10/7/30%, min/avg/max = 6.58/169/1037  
Fri Feb 6 02:51:11 UTC 2009  
^Cmbenito@liendre:~/Desktop\$

## RETARDO = 6 segundos

mбенито@liendre:~/Desktop\$ sh arranque  
Fri Feb 6 02:50:12 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:13 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:14 UTC 2009  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
10.10.0.10 : [3], 84 bytes, 65.8 ms (65.8 avg, 75% loss)  
10.10.0.10 : [4], 84 bytes, 23.1 ms (44.5 avg, 60% loss)  
10.10.0.10 : [5], 84 bytes, 12.2 ms (33.7 avg, 50% loss)  
10.10.0.10 : [6], 84 bytes, 19.0 ms (30.0 avg, 42% loss)  
10.10.0.10 : [7], 84 bytes, 4.70 ms (25.0 avg, 37% loss)  
10.10.0.10 : [8], 84 bytes, 20.4 ms (24.2 avg, 33% loss)  
10.10.0.10 : [9], 84 bytes, 5.16 ms (21.5 avg, 30% loss)

10.10.0.10 : xmt/rcv/%loss = 10/7/30%, min/avg/max = 4.70/21.5/65.8  
Fri Feb 6 02:50:24 UTC 2009  
^Cmbenito@liendre:~/Desktop\$

## RETARDO = 6 segundos

mбенито@liendre:~/Desktop\$ sh arranque  
Fri Feb 6 02:50:00 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:01 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:02 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:03 UTC 2009

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
10.10.0.10 : [1], 84 bytes, 17.2 ms (17.2 avg, 50% loss)  
10.10.0.10 : [2], 84 bytes, 29.7 ms (23.4 avg, 33% loss)  
10.10.0.10 : [3], 84 bytes, 20.7 ms (22.5 avg, 25% loss)  
10.10.0.10 : [4], 84 bytes, 10.2 ms (19.4 avg, 20% loss)  
10.10.0.10 : [8], 84 bytes, 10.5 ms (17.7 avg, 44% loss)  
10.10.0.10 : [9], 84 bytes, 12.4 ms (16.8 avg, 40% loss)

10.10.0.10 : xmt/rcv/%loss = 10/6/40%, min/avg/max = 10.2/16.8/29.7  
Fri Feb 6 02:50:12 UTC 2009

**^Cmbenito@liendre:~/Desktop\$**

## RETARDO = 4 segundos

**mbenito@liendre:~/Desktop\$ sh arranque**  
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:55 UTC 2009  
10.10.0.10 : [0], 84 bytes, 9.44 ms (9.44 avg, 0% loss)  
10.10.0.10 : [1], 84 bytes, 417 ms (213 avg, 0% loss)  
10.10.0.10 : [2], 84 bytes, 45.2 ms (157 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 13.5 ms (121 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 10.1 ms (99.2 avg, 0% loss)  
10.10.0.10 : [5], 84 bytes, 13.2 ms (84.9 avg, 0% loss)  
10.10.0.10 : [6], 84 bytes, 4.72 ms (73.4 avg, 0% loss)  
10.10.0.10 : [7], 84 bytes, 14.2 ms (66.0 avg, 0% loss)  
10.10.0.10 : [8], 84 bytes, 17.1 ms (60.6 avg, 0% loss)  
10.10.0.10 : [9], 84 bytes, 18.7 ms (56.4 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 4.72/56.4/417  
Fri Feb 6 02:50:14 UTC 2009

**^Cmbenito@liendre:~/Desktop\$**

## RETARDO = 2 segundos

**mbenito@liendre:~/Desktop\$ sh arranque**  
Fri Feb 6 02:49:50 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:51 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:57 UTC 2009  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

```

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [0], 84 bytes, 54.4 ms (54.4 avg, 0% loss)
10.10.0.10 : [1], 84 bytes, 4.41 ms (29.4 avg, 0% loss)
10.10.0.10 : [2], 84 bytes, 10.5 ms (23.1 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 5.32 ms (18.6 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 17.8 ms (18.5 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 98.1 ms (31.7 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 34.6 ms (32.1 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 45.2 ms (33.8 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 14.0 ms (31.6 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 6.99 ms (29.1 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 4.41/29.1/98.1
mbenito@liendre:~/Desktop$

```

## RETARDO = 7 segundos

```

mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:50:04 UTC 2009

```

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:05 UTC 2009

```

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:06 UTC 2009

```

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:07 UTC 2009

```

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:08 UTC 2009

```

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:09 UTC 2009

```

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:10 UTC 2009
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [0], 84 bytes, 1039 ms (1039 avg, 50% loss)
10.10.0.10 : [1], 84 bytes, 54.4 ms (546 avg, 0% loss)
10.10.0.10 : [2], 84 bytes, 11.3 ms (368 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 7.92 ms (278 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 8.06 ms (224 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 459 ms (263 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 11.9 ms (227 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 4.57 ms (199 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 7.53 ms (178 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 14.4 ms (161 avg, 0% loss)
mbenito@liendre:~/Desktop$

```

## RETARDO = 6 segundos

```

mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:47 UTC 2009

```

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:48 UTC 2009

```

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:49 UTC 2009

```

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:50 UTC 2009

```

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:51 UTC 2009
10.10.0.10 : [0], 84 bytes, 17.0 ms (17.0 avg, 0% loss)
10.10.0.10 : [1], 84 bytes, 22.2 ms (19.6 avg, 0% loss)

```

```

10.10.0.10 : [2], 84 bytes, 15.1 ms (18.1 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 32.4 ms (21.7 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 24.1 ms (22.2 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 7.59 ms (19.7 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 128 ms (35.3 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 5.72 ms (31.6 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 18.5 ms (30.1 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 17.7 ms (28.9 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 5.72/28.9/128
Fri Feb 6 02:50:00 UTC 2009
10.10.0.10 : [0], 84 bytes, 9.85 ms (9.85 avg, 0% loss)
10.10.0.10 : [1], 84 bytes, 49.4 ms (29.6 avg, 0% loss)
10.10.0.10 : [2], 84 bytes, 237 ms (99.0 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 8.85 ms (76.4 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 7.94 ms (62.7 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 33.5 ms (57.9 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 2010 ms (336 avg, 12% loss)
10.10.0.10 : [7], 84 bytes, 1001 ms (419 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 10.8 ms (374 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 158 ms (352 avg, 0% loss)

```

```

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 7.94/352/2010
Fri Feb 6 02:50:09 UTC 2009
^Cmbenito@liendre:~/Desktop$

```

## RETARDO = 4 segundos

```

mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:50:10 UTC 2009

```

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:11 UTC 2009

```

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:12 UTC 2009

```

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:13 UTC 2009

```

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:14 UTC 2009

```

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:15 UTC 2009
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [0], 84 bytes, 2039 ms (2039 avg, 66% loss)
10.10.0.10 : [1], 84 bytes, 1031 ms (1535 avg, 33% loss)
10.10.0.10 : [2], 84 bytes, 33.7 ms (1035 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 1014 ms (1029 avg, 20% loss)
10.10.0.10 : [4], 84 bytes, 17.1 ms (827 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 28.6 ms (694 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 28.9 ms (599 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 13.4 ms (525 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 87.1 ms (477 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 11.8 ms (430 avg, 0% loss)

```

```

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 11.8/430/2039
Fri Feb 6 02:50:24 UTC 2009
10.10.0.10 : [0], 84 bytes, 8.73 ms (8.73 avg, 0% loss)
10.10.0.10 : [1], 84 bytes, 3012 ms (1510 avg, 50% loss)
10.10.0.10 : [2], 84 bytes, 2013 ms (1678 avg, 25% loss)
10.10.0.10 : [3], 84 bytes, 1004 ms (1509 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 9.00 ms (1209 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 10.4 ms (1009 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 66.8 ms (875 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 5.03 ms (766 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 14.6 ms (682 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 52.2 ms (619 avg, 0% loss)

```

```

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 5.03/619/3012
Fri Feb 6 02:50:33 UTC 2009
^Cmbenito@liendre:~/Desktop$

```

## RETARDO = 5 segundos

**mбенито@liendre:~/Desktop\$ sh arranque**

Fri Feb 6 02:49:50 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:51 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:55 UTC 2009

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

10.10.0.10 : [0], 84 bytes, 1046 ms (1046 avg, 50% loss)

10.10.0.10 : [1], 84 bytes, 38.0 ms (542 avg, 0% loss)

10.10.0.10 : [2], 84 bytes, 25.0 ms (369 avg, 0% loss)

10.10.0.10 : [3], 84 bytes, 12.1 ms (280 avg, 0% loss)

10.10.0.10 : [4], 84 bytes, 6.88 ms (225 avg, 0% loss)

10.10.0.10 : [5], 84 bytes, 15.1 ms (190 avg, 0% loss)

10.10.0.10 : [6], 84 bytes, 11.5 ms (165 avg, 0% loss)

10.10.0.10 : [7], 84 bytes, 8.37 ms (145 avg, 0% loss)

10.10.0.10 : [8], 84 bytes, 5.48 ms (129 avg, 0% loss)

10.10.0.10 : [9], 84 bytes, 42.4 ms (121 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 5.48/121/1046

Fri Feb 6 02:50:04 UTC 2009

10.10.0.10 : [0], 84 bytes, 9.65 ms (9.65 avg, 0% loss)

10.10.0.10 : [1], 84 bytes, 6.90 ms (8.27 avg, 0% loss)

10.10.0.10 : [2], 84 bytes, 92.9 ms (36.5 avg, 0% loss)

10.10.0.10 : [3], 84 bytes, 5.91 ms (28.8 avg, 0% loss)

10.10.0.10 : [4], 84 bytes, 8.48 ms (24.7 avg, 0% loss)

10.10.0.10 : [5], 84 bytes, 15.3 ms (23.2 avg, 0% loss)

10.10.0.10 : [6], 84 bytes, 9.98 ms (21.3 avg, 0% loss)

10.10.0.10 : [7], 84 bytes, 120 ms (33.7 avg, 0% loss)

10.10.0.10 : [8], 84 bytes, 12.4 ms (31.3 avg, 0% loss)

10.10.0.10 : [9], 84 bytes, 12.0 ms (29.4 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 5.91/29.4/120

Fri Feb 6 02:50:13 UTC 2009

**^Cmбенито@liendre:~/Desktop\$**

## RETARDO = 4 segundos

**mбенито@liendre:~/Desktop\$ sh arranque**

Fri Feb 6 02:49:50 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:51 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : [0], 84 bytes, 2028 ms (2028 avg, 66% loss)

10.10.0.10 : [1], 84 bytes, 1029 ms (1528 avg, 33% loss)

10.10.0.10 : [2], 84 bytes, 78.7 ms (1045 avg, 0% loss)

10.10.0.10 : [3], 84 bytes, 2040 ms (1294 avg, 33% loss)

10.10.0.10 : [4], 84 bytes, 1032 ms (1241 avg, 16% loss)

10.10.0.10 : [5], 84 bytes, 32.3 ms (1040 avg, 0% loss)

10.10.0.10 : [6], 84 bytes, 31.9 ms (896 avg, 0% loss)

10.10.0.10 : [7], 84 bytes, 6.27 ms (784 avg, 0% loss)

10.10.0.10 : [8], 84 bytes, 13.8 ms (699 avg, 0% loss)

10.10.0.10 : [9], 84 bytes, 192 ms (648 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 6.27/648/2040  
Fri Feb 6 02:50:03 UTC 2009  
^Cmbenito@liendre:~/Desktop\$

## RETARDO = 4 segundos

mbenito@liendre:~/Desktop\$ sh arranque  
Fri Feb 6 02:49:48 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:49 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:50 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:51 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:55 UTC 2009  
10.10.0.10 : [0], 84 bytes, 1018 ms (1018 avg, 50% loss)  
10.10.0.10 : [1], 84 bytes, 17.6 ms (518 avg, 0% loss)  
10.10.0.10 : [2], 84 bytes, 30.5 ms (355 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 6.67 ms (268 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 2015 ms (617 avg, 16% loss)  
10.10.0.10 : [5], 84 bytes, 1007 ms (682 avg, 0% loss)  
10.10.0.10 : [6], 84 bytes, 25.2 ms (588 avg, 0% loss)  
10.10.0.10 : [7], 84 bytes, 40.4 ms (520 avg, 0% loss)  
10.10.0.10 : [8], 84 bytes, 7.62 ms (463 avg, 0% loss)  
10.10.0.10 : [9], 84 bytes, 9.62 ms (417 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 6.67/417/2015  
Fri Feb 6 02:50:04 UTC 2009  
^Cmbenito@liendre:~/Desktop\$

## RETARDO = 7 segundos

mbenito@liendre:~/Desktop\$ sh arranque  
Fri Feb 6 02:49:49 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:50 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:51 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:54 UTC 2009  
10.10.0.10 : [0], 84 bytes, 35.7 ms (35.7 avg, 0% loss)  
10.10.0.10 : [1], 84 bytes, 6.01 ms (20.9 avg, 0% loss)  
10.10.0.10 : [2], 84 bytes, 31.2 ms (24.3 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 6.64 ms (19.9 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 33.9 ms (22.7 avg, 0% loss)  
10.10.0.10 : [5], 84 bytes, 8.83 ms (20.4 avg, 0% loss)  
10.10.0.10 : [6], 84 bytes, 19.8 ms (20.3 avg, 0% loss)  
10.10.0.10 : [7], 84 bytes, 99.3 ms (30.2 avg, 0% loss)  
10.10.0.10 : [8], 84 bytes, 24.1 ms (29.5 avg, 0% loss)  
10.10.0.10 : [9], 84 bytes, 23.1 ms (28.9 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 6.01/28.9/99.3  
Fri Feb 6 02:50:03 UTC 2009  
^Cmbenito@liendre:~/Desktop\$

## RETARDO = 5 segundos

mbenito@liendre:~/Desktop\$ sh arranque  
Fri Feb 6 02:50:09 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:10 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:11 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:12 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:13 UTC 2009  
10.10.0.10 : [0], 84 bytes, 2076 ms (2076 avg, 66% loss)  
10.10.0.10 : [1], 84 bytes, 1072 ms (1574 avg, 33% loss)  
10.10.0.10 : [2], 84 bytes, 72.1 ms (1073 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 7.73 ms (807 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 18.9 ms (649 avg, 0% loss)  
10.10.0.10 : [5], 84 bytes, 29.2 ms (546 avg, 0% loss)  
10.10.0.10 : [6], 84 bytes, 305 ms (511 avg, 0% loss)  
10.10.0.10 : [7], 84 bytes, 45.5 ms (453 avg, 0% loss)  
10.10.0.10 : [8], 84 bytes, 108 ms (415 avg, 0% loss)  
10.10.0.10 : [9], 84 bytes, 24.1 ms (376 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 7.73/376/2076  
Fri Feb 6 02:50:22 UTC 2009  
^Cmbenito@liendre:~/Desktop\$

## RETARDO = 4 segundos

mbenito@liendre:~/Desktop\$ sh arranque  
Fri Feb 6 02:49:49 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:50 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:51 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:55 UTC 2009  
10.10.0.10 : [0], 84 bytes, 1039 ms (1039 avg, 50% loss)  
10.10.0.10 : [1], 84 bytes, 37.7 ms (538 avg, 0% loss)  
10.10.0.10 : [2], 84 bytes, 180 ms (419 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 100 ms (339 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 10.1 ms (273 avg, 0% loss)  
10.10.0.10 : [5], 84 bytes, 34.2 ms (233 avg, 0% loss)  
10.10.0.10 : [6], 84 bytes, 24.6 ms (203 avg, 0% loss)  
10.10.0.10 : [7], 84 bytes, 18.9 ms (180 avg, 0% loss)  
10.10.0.10 : [8], 84 bytes, 19.4 ms (162 avg, 0% loss)  
10.10.0.10 : [9], 84 bytes, 22.2 ms (148 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 10.1/148/1039  
Fri Feb 6 02:50:04 UTC 2009  
^Cmbenito@liendre:~/Desktop\$

## RETARDO = 6 segundos



**mбенито@liendre:~/Desktop\$ sh arranque**

Fri Feb 6 02:49:59 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:00 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:01 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:02 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:03 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:04 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:05 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:06 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:07 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:08 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:09 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:10 UTC 2009

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

10.10.0.10 : [0], 84 bytes, 2033 ms (2033 avg, 66% loss)

10.10.0.10 : [1], 84 bytes, 1024 ms (1528 avg, 33% loss)

10.10.0.10 : [2], 84 bytes, 14.5 ms (1023 avg, 0% loss)

10.10.0.10 : [3], 84 bytes, 22.5 ms (773 avg, 0% loss)

10.10.0.10 : [4], 84 bytes, 10.6 ms (621 avg, 0% loss)

10.10.0.10 : [5], 84 bytes, 21.6 ms (521 avg, 0% loss)

10.10.0.10 : [6], 84 bytes, 77.0 ms (457 avg, 0% loss)

10.10.0.10 : [7], 84 bytes, 47.5 ms (406 avg, 0% loss)

10.10.0.10 : [8], 84 bytes, 23.7 ms (363 avg, 0% loss)

10.10.0.10 : [9], 84 bytes, 37.8 ms (331 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 10.6/331/2033

Fri Feb 6 02:50:19 UTC 2009

**^Cmбенито@liendre:~/Desktop\$**

## RETARDO = 11 segundos

**mбенито@liendre:~/Desktop\$ sh arranque**

Fri Feb 6 02:50:01 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:02 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:03 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:04 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:05 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:06 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:07 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

```
Fri Feb 6 02:50:08 UTC 2009
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [0], 84 bytes, 1025 ms (1025 avg, 50% loss)
10.10.0.10 : [1], 84 bytes, 23.9 ms (524 avg, 0% loss)
10.10.0.10 : [2], 84 bytes, 5.04 ms (351 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 16.6 ms (267 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 60.4 ms (226 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 1047 ms (363 avg, 14% loss)
10.10.0.10 : [6], 84 bytes, 41.1 ms (317 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 12.5 ms (279 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 16.1 ms (249 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 12.2 ms (226 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 5.04/226/1047
Fri Feb 6 02:50:17 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 7 segundos

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:49 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:50 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:51 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:52 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:53 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:54 UTC 2009
10.10.0.10 : [0], 84 bytes, 1161 ms (1161 avg, 50% loss)
10.10.0.10 : [1], 84 bytes, 159 ms (660 avg, 0% loss)
10.10.0.10 : [2], 84 bytes, 67.7 ms (463 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 60.5 ms (362 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 35.7 ms (297 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 12.7 ms (249 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 54.0 ms (221 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 31.3 ms (197 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 26.2 ms (178 avg, 10% loss)
```

```
10.10.0.10 : xmt/rcv/%loss = 10/9/10%, min/avg/max = 12.7/178/1161
Fri Feb 6 02:50:03 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 5 segundos

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:48 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:49 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:50 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:51 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:52 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
```

Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:57 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:58 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:59 UTC 2009

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

10.10.0.10 : [0], 84 bytes, 2027 ms (2027 avg, 66% loss)

10.10.0.10 : [1], 84 bytes, 1022 ms (1524 avg, 33% loss)

10.10.0.10 : [2], 84 bytes, 12.4 ms (1020 avg, 0% loss)

10.10.0.10 : [3], 84 bytes, 38.2 ms (774 avg, 0% loss)

10.10.0.10 : [4], 84 bytes, 8.08 ms (621 avg, 0% loss)

10.10.0.10 : [5], 84 bytes, 77.6 ms (530 avg, 0% loss)

10.10.0.10 : [6], 84 bytes, 10.6 ms (456 avg, 0% loss)

10.10.0.10 : [7], 84 bytes, 9.71 ms (400 avg, 0% loss)

10.10.0.10 : [8], 84 bytes, 8.87 ms (357 avg, 0% loss)

10.10.0.10 : [9], 84 bytes, 48.4 ms (326 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 8.08/326/2027

Fri Feb 6 02:50:08 UTC 2009

**^Cmbenito@liendre:~/Desktop\$**

## RETARDO = 11 segundos

**mbenito@liendre:~/Desktop\$ sh arranque**

Fri Feb 6 02:49:51 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : [0], 84 bytes, 2065 ms (2065 avg, 66% loss)

10.10.0.10 : [1], 84 bytes, 1063 ms (1564 avg, 33% loss)

10.10.0.10 : [2], 84 bytes, 72.6 ms (1067 avg, 0% loss)

10.10.0.10 : [3], 84 bytes, 17.1 ms (804 avg, 0% loss)

10.10.0.10 : [4], 84 bytes, 5.52 ms (644 avg, 0% loss)

10.10.0.10 : [5], 84 bytes, 30.1 ms (542 avg, 0% loss)

10.10.0.10 : [6], 84 bytes, 13.1 ms (466 avg, 0% loss)

10.10.0.10 : [7], 84 bytes, 9.37 ms (409 avg, 0% loss)

10.10.0.10 : [8], 84 bytes, 26.5 ms (367 avg, 0% loss)

10.10.0.10 : [9], 84 bytes, 15.7 ms (331 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 5.52/331/2065

Fri Feb 6 02:50:05 UTC 2009

**^Cmbenito@liendre:~/Desktop\$**

## RETARDO = 4 segundos

**mbenito@liendre:~/Desktop\$ sh arranque**

Fri Feb 6 02:49:47 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:48 UTC 2009

```

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:49 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:50 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:51 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:57 UTC 2009
10.10.0.10 : [0], 84 bytes, 2213 ms (2213 avg, 66% loss)
10.10.0.10 : [1], 84 bytes, 1231 ms (1722 avg, 33% loss)
10.10.0.10 : [2], 84 bytes, 222 ms (1222 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 11.4 ms (919 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 9.83 ms (737 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 50.6 ms (623 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 2047 ms (826 avg, 22% loss)
10.10.0.10 : [7], 84 bytes, 1047 ms (854 avg, 11% loss)
10.10.0.10 : [8], 84 bytes, 43.6 ms (764 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 14.6 ms (689 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 9.83/689/2213
Fri Feb 6 02:50:06 UTC 2009
^Cmbenito@liendre:~/Desktop$

```

## RETARDO = 10 segundos

```

mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:47 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:48 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:49 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:50 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:51 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:54 UTC 2009
ICMP Host Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.10
ICMP Host Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.10
ICMP Host Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [3], 84 bytes, 55.4 ms (55.4 avg, 75% loss)
10.10.0.10 : [4], 84 bytes, 15.7 ms (35.6 avg, 60% loss)
10.10.0.10 : [5], 84 bytes, 77.0 ms (49.4 avg, 50% loss)
10.10.0.10 : [6], 84 bytes, 83.7 ms (58.0 avg, 42% loss)
10.10.0.10 : [7], 84 bytes, 11.1 ms (48.6 avg, 37% loss)
10.10.0.10 : [8], 84 bytes, 8.17 ms (41.8 avg, 33% loss)

```

```
10.10.0.10 : [9], 84 bytes, 22.9 ms (39.1 avg, 30% loss)

10.10.0.10 : xmt/rcv/%loss = 10/7/30%, min/avg/max = 8.17/39.1/83.7
Fri Feb 6 02:50:04 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 11 segundos

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:47 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:48 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:49 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:50 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:51 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:53 UTC 2009
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [2], 84 bytes, 2037 ms (2037 avg, 80% loss)
10.10.0.10 : [3], 84 bytes, 1028 ms (1532 avg, 60% loss)
10.10.0.10 : [4], 84 bytes, 41.0 ms (1035 avg, 40% loss)
10.10.0.10 : [5], 84 bytes, 25.0 ms (782 avg, 33% loss)
10.10.0.10 : [6], 84 bytes, 13.3 ms (629 avg, 28% loss)
10.10.0.10 : [7], 84 bytes, 11.0 ms (526 avg, 25% loss)

10.10.0.10 : xmt/rcv/%loss = 10/6/40%, min/avg/max = 11.0/526/2037
Fri Feb 6 02:50:03 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 9 segundos

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:47 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:48 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:49 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:50 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:51 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:55 UTC 2009
10.10.0.10 : [0], 84 bytes, 50.7 ms (50.7 avg, 0% loss)
10.10.0.10 : [1], 84 bytes, 12.3 ms (31.5 avg, 0% loss)
10.10.0.10 : [2], 84 bytes, 28.3 ms (30.4 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 16.6 ms (27.0 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 22.1 ms (26.0 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 15.8 ms (24.3 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 16.3 ms (23.1 avg, 0% loss)
```

```
10.10.0.10 : [7], 84 bytes, 10.7 ms (21.6 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 277 ms (50.0 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 5.76 ms (45.6 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 5.76/45.6/277
Fri Feb 6 02:50:04 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 8 segundos

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:48 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:49 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:50 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:51 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:53 UTC 2009
10.10.0.10 : [0], 84 bytes, 2009 ms (2009 avg, 50% loss)
10.10.0.10 : [1], 84 bytes, 1003 ms (1506 avg, 0% loss)
10.10.0.10 : [2], 84 bytes, 25.4 ms (1012 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 11.0 ms (762 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 57.3 ms (621 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 11.0 ms (519 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 8.21 ms (446 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 26.4 ms (394 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 21.2 ms (352 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 14.1 ms (318 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 8.21/318/2009
Fri Feb 6 02:50:02 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 5 segundos

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:47 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:48 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:49 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:50 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:51 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
```

```
Fri Feb 6 02:49:57 UTC 2009
10.10.0.10 : [0], 84 bytes, 1125 ms (1125 avg, 50% loss)
10.10.0.10 : [1], 84 bytes, 118 ms (622 avg, 0% loss)
10.10.0.10 : [2], 84 bytes, 7.04 ms (417 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 11.2 ms (315 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 18.0 ms (256 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 22.8 ms (217 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 12.7 ms (188 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 26.3 ms (167 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 19.8 ms (151 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 71.5 ms (143 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 7.04/143/1125
Fri Feb 6 02:50:06 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 10 segundos

```
mbenito@liendre:~/Desktop$ sh arranque
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:57 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:58 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:59 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:00 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:01 UTC 2009
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [3], 84 bytes, 3037 ms (3037 avg, 85% loss)
10.10.0.10 : [4], 84 bytes, 2040 ms (2538 avg, 71% loss)
10.10.0.10 : [5], 84 bytes, 1031 ms (2036 avg, 57% loss)
10.10.0.10 : [6], 84 bytes, 29.3 ms (1534 avg, 42% loss)
10.10.0.10 : [7], 84 bytes, 7.57 ms (1229 avg, 37% loss)
10.10.0.10 : [8], 84 bytes, 11.7 ms (1026 avg, 33% loss)

10.10.0.10 : xmt/rcv/%loss = 10/6/40%, min/avg/max = 7.57/1026/3037
Fri Feb 6 02:50:10 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 12 segundos

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:55 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:57 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:58 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:00 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:01 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:02 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:03 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:04 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:05 UTC 2009
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [0], 84 bytes, 35.8 ms (35.8 avg, 0% loss)
10.10.0.10 : [1], 84 bytes, 105 ms (70.4 avg, 0% loss)
10.10.0.10 : [2], 84 bytes, 1034 ms (391 avg, 25% loss)
10.10.0.10 : [3], 84 bytes, 25.4 ms (300 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 100 ms (260 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 170 ms (245 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/6/40%, min/avg/max = 25.4/245/1034
Fri Feb 6 02:50:15 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

**RETARDO = 13 segundos**



## A2.6. MEDIDAS DEL RETARDO CUANDO CAE UN ENLACE

A continuación mostramos los resultados del tiempo que tarda un nodo en detectar la caída de un enlace en un sentido que esta siendo utilizado y evaluar otro enlace para continuar la comunicación.

[illegible]

10.10.0.8 : [70], 84 bytes, 31.9 ms (65.9 avg, 67% loss)  
 10.10.0.8 : [71], 84 bytes, 235 ms (73.0 avg, 66% loss)  
 10.10.0.8 : [72], 84 bytes, 20.2 ms (70.9 avg, 65% loss)  
 10.10.0.8 : [73], 84 bytes, 6.17 ms (68.4 avg, 64% loss)  
 10.10.0.8 : [74], 84 bytes, 10.0 ms (66.2 avg, 64% loss)  
 10.10.0.8 : [75], 84 bytes, 5.09 ms (64.0 avg, 63% loss)  
 10.10.0.8 : [76], 84 bytes, 15.3 ms (62.3 avg, 62% loss)  
 10.10.0.8 : [77], 84 bytes, 4.89 ms (60.4 avg, 61% loss)  
 10.10.0.8 : [78], 84 bytes, 3.73 ms (58.6 avg, 60% loss)  
 10.10.0.8 : [79], 84 bytes, 3.83 ms (56.9 avg, 60% loss)  
 10.10.0.8 : [80], 84 bytes, 147 ms (59.6 avg, 59% loss)  
 10.10.0.8 : [81], 84 bytes, 78.4 ms (60.2 avg, 58% loss)  
 10.10.0.8 : [82], 84 bytes, 15.5 ms (58.9 avg, 57% loss)  
 10.10.0.8 : [83], 84 bytes, 10.4 ms (57.6 avg, 57% loss)  
 10.10.0.8 : [84], 84 bytes, 29.8 ms (56.8 avg, 56% loss)  
 10.10.0.8 : [85], 84 bytes, 18.6 ms (55.8 avg, 55% loss)  
 10.10.0.8 : [86], 84 bytes, 34.1 ms (55.2 avg, 55% loss)  
 10.10.0.8 : [87], 84 bytes, 6.32 ms (54.0 avg, 54% loss)  
 10.10.0.8 : [88], 84 bytes, 12.1 ms (53.0 avg, 53% loss)  
 10.10.0.8 : [89], 84 bytes, 9.11 ms (51.9 avg, 53% loss)  
 10.10.0.8 : [90], 84 bytes, 13.5 ms (51.1 avg, 52% loss)  
 10.10.0.8 : [91], 84 bytes, 8.49 ms (50.1 avg, 52% loss)  
 10.10.0.8 : [92], 84 bytes, 15.3 ms (49.3 avg, 51% loss)  
 10.10.0.8 : [93], 84 bytes, 80.9 ms (50.0 avg, 51% loss)  
 10.10.0.8 : [94], 84 bytes, 71.8 ms (50.5 avg, 50% loss)  
 10.10.0.8 : [95], 84 bytes, 576 ms (61.4 avg, 50% loss)  
 10.10.0.8 : [96], 84 bytes, 7.48 ms (60.3 avg, 49% loss)  
 10.10.0.8 : [97], 84 bytes, 105 ms (61.2 avg, 48% loss)  
 10.10.0.8 : [98], 84 bytes, 7.70 ms (60.2 avg, 48% loss)  
 10.10.0.8 : [99], 84 bytes, 7.91 ms (59.2 avg, 48% loss)  
 10.10.0.8 : [100], 84 bytes, 43.2 ms (58.9 avg, 47% loss)  
 10.10.0.8 : [101], 84 bytes, 8.79 ms (57.9 avg, 47% loss)  
 10.10.0.8 : [102], 84 bytes, 8.32 ms (57.0 avg, 46% loss)  
 10.10.0.8 : [103], 84 bytes, 11.4 ms (56.2 avg, 46% loss)  
 10.10.0.8 : [104], 84 bytes, 108 ms (57.1 avg, 45% loss)  
 10.10.0.8 : [105], 84 bytes, 7.49 ms (56.3 avg, 45% loss)  
 10.10.0.8 : [106], 84 bytes, 7.29 ms (55.4 avg, 44% loss)  
 10.10.0.8 : [107], 84 bytes, 22.3 ms (54.9 avg, 44% loss)  
 10.10.0.8 : [108], 84 bytes, 80.5 ms (55.3 avg, 44% loss)  
 10.10.0.8 : [109], 84 bytes, 16.2 ms (54.7 avg, 43% loss)  
 10.10.0.8 : [110], 84 bytes, 15.5 ms (54.1 avg, 43% loss)  
 10.10.0.8 : [111], 84 bytes, 19.2 ms (53.5 avg, 42% loss)  
 10.10.0.8 : [112], 84 bytes, 6.71 ms (52.8 avg, 42% loss)  
 10.10.0.8 : [113], 84 bytes, 6.01 ms (52.1 avg, 42% loss)  
 10.10.0.8 : [114], 84 bytes, 26.1 ms (51.7 avg, 41% loss)  
 10.10.0.8 : [115], 84 bytes, 9.72 ms (51.1 avg, 41% loss)  
 10.10.0.8 : [116], 84 bytes, 38.3 ms (50.9 avg, 41% loss)  
 10.10.0.8 : [117], 84 bytes, 14.0 ms (50.4 avg, 40% loss)  
 10.10.0.8 : [118], 84 bytes, 54.6 ms (50.4 avg, 40% loss)  
 10.10.0.8 : [119], 84 bytes, 376 ms (55.0 avg, 40% loss)  
 10.10.0.8 : [120], 84 bytes, 35.5 ms (54.7 avg, 39% loss)  
 10.10.0.8 : [121], 84 bytes, 4.53 ms (54.0 avg, 39% loss)  
 10.10.0.8 : [122], 84 bytes, 57.8 ms (54.1 avg, 39% loss)  
 10.10.0.8 : [123], 84 bytes, 23.6 ms (53.7 avg, 38% loss)  
 10.10.0.8 : [124], 84 bytes, 6.02 ms (53.0 avg, 38% loss)  
 10.10.0.8 : [125], 84 bytes, 3.56 ms (52.4 avg, 38% loss)  
 10.10.0.8 : [126], 84 bytes, 7.98 ms (51.8 avg, 37% loss)  
 10.10.0.8 : [127], 84 bytes, 445 ms (56.8 avg, 37% loss)  
 10.10.0.8 : [128], 84 bytes, 44.2 ms (56.6 avg, 37% loss)  
 10.10.0.8 : [129], 84 bytes, 8.85 ms (56.0 avg, 36% loss)  
 10.10.0.8 : [130], 84 bytes, 27.5 ms (55.7 avg, 36% loss)  
 10.10.0.8 : [131], 84 bytes, 22.6 ms (55.3 avg, 36% loss)  
 10.10.0.8 : [132], 84 bytes, 92.8 ms (55.7 avg, 36% loss)  
 10.10.0.8 : [133], 84 bytes, 33.3 ms (55.5 avg, 35% loss)  
 10.10.0.8 : [134], 84 bytes, 40.4 ms (55.3 avg, 35% loss)  
 10.10.0.8 : [135], 84 bytes, 95.5 ms (55.7 avg, 35% loss)  
 10.10.0.8 : [136], 84 bytes, 6.18 ms (55.2 avg, 35% loss)  
 10.10.0.8 : [137], 84 bytes, 478 ms (59.9 avg, 34% loss)  
 10.10.0.8 : [138], 84 bytes, 543 ms (65.2 avg, 34% loss)  
 10.10.0.8 : [139], 84 bytes, 12.3 ms (64.6 avg, 34% loss)  
 10.10.0.8 : [140], 84 bytes, 88.5 ms (64.9 avg, 34% loss)  
 10.10.0.8 : [141], 84 bytes, 15.5 ms (64.4 avg, 33% loss)

10.10.0.8 : xmt/rcv/%loss = 142/94/33%, min/avg/max = 3.46/64.4/798  
 root@OpenWrt:~#

## RETARDO: 49 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.8 -i 1 -c 150
10.10.0.8 : [0], 84 bytes, 1014 ms (1014 avg, 50% loss)
10.10.0.8 : [1], 84 bytes, 8.91 ms (511 avg, 0% loss)
10.10.0.8 : [2], 84 bytes, 55.6 ms (359 avg, 0% loss)
10.10.0.8 : [3], 84 bytes, 9.37 ms (272 avg, 0% loss)
10.10.0.8 : [4], 84 bytes, 2.01 ms (218 avg, 0% loss)
10.10.0.8 : [5], 84 bytes, 2.60 ms (182 avg, 0% loss)
10.10.0.8 : [6], 84 bytes, 1.59 ms (156 avg, 0% loss)
10.10.0.8 : [7], 84 bytes, 4.25 ms (137 avg, 0% loss)
10.10.0.8 : [8], 84 bytes, 2.37 ms (122 avg, 0% loss)
10.10.0.8 : [9], 84 bytes, 1.98 ms (110 avg, 0% loss)
10.10.0.8 : [10], 84 bytes, 6.28 ms (100 avg, 0% loss)
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
10.10.0.8 : [49], 84 bytes, 7.87 ms (93.1 avg, 76% loss)
10.10.0.8 : [50], 84 bytes, 11.2 ms (86.8 avg, 74% loss)
10.10.0.8 : [51], 84 bytes, 13.6 ms (81.5 avg, 73% loss)
10.10.0.8 : [52], 84 bytes, 5.14 ms (76.4 avg, 71% loss)
10.10.0.8 : [53], 84 bytes, 18.9 ms (72.8 avg, 70% loss)
10.10.0.8 : [54], 84 bytes, 20.4 ms (69.8 avg, 69% loss)
10.10.0.8 : [55], 84 bytes, 25.2 ms (67.3 avg, 67% loss)
10.10.0.8 : [56], 84 bytes, 16.1 ms (64.6 avg, 66% loss)
10.10.0.8 : [57], 84 bytes, 7.88 ms (61.8 avg, 65% loss)
10.10.0.8 : [58], 84 bytes, 11.7 ms (59.4 avg, 64% loss)
10.10.0.8 : [59], 84 bytes, 17.0 ms (57.4 avg, 63% loss)
10.10.0.8 : [60], 84 bytes, 25.5 ms (56.1 avg, 62% loss)
10.10.0.8 : [61], 84 bytes, 6.80 ms (54.0 avg, 61% loss)
10.10.0.8 : [62], 84 bytes, 4.85 ms (52.0 avg, 60% loss)
10.10.0.8 : [63], 84 bytes, 101 ms (53.9 avg, 59% loss)
10.10.0.8 : [64], 84 bytes, 9.19 ms (52.3 avg, 58% loss)
10.10.0.8 : [65], 84 bytes, 5.21 ms (50.6 avg, 57% loss)
10.10.0.8 : [66], 84 bytes, 29.3 ms (49.8 avg, 56% loss)
10.10.0.8 : [67], 84 bytes, 13.8 ms (48.4 avg, 55% loss)
10.10.0.8 : [68], 84 bytes, 20.8 ms (47.7 avg, 55% loss)
10.10.0.8 : [69], 84 bytes, 12.7 ms (46.6 avg, 54% loss)
10.10.0.8 : [70], 84 bytes, 8.61 ms (45.5 avg, 53% loss)
10.10.0.8 : [71], 84 bytes, 6.23 ms (44.3 avg, 52% loss)
10.10.0.8 : [72], 84 bytes, 12.0 ms (43.4 avg, 52% loss)
10.10.0.8 : [73], 84 bytes, 72.0 ms (44.2 avg, 51% loss)
10.10.0.8 : [74], 84 bytes, 149 ms (47.0 avg, 50% loss)
10.10.0.8 : [75], 84 bytes, 12.7 ms (46.1 avg, 50% loss)
10.10.0.8 : [76], 84 bytes, 6.01 ms (45.1 avg, 49% loss)
10.10.0.8 : [77], 84 bytes, 91.1 ms (46.3 avg, 48% loss)
10.10.0.8 : [78], 84 bytes, 17.9 ms (45.6 avg, 48% loss)
10.10.0.8 : [79], 84 bytes, 50.7 ms (45.7 avg, 47% loss)
10.10.0.8 : [80], 84 bytes, 14.5 ms (45.0 avg, 46% loss)
10.10.0.8 : [81], 84 bytes, 7.63 ms (44.1 avg, 46% loss)
10.10.0.8 : [82], 84 bytes, 21.3 ms (43.6 avg, 45% loss)
10.10.0.8 : [83], 84 bytes, 14.0 ms (43.0 avg, 45% loss)
10.10.0.8 : [84], 84 bytes, 12.5 ms (42.3 avg, 44% loss)
```

```
10.10.0.8 : xmt/rcv/%loss = 100/61/39%, min/avg/max = 1.59/39.1/1014
root@OpenWrt:~#
```

```
root@OpenWrt:~# fping 10.10.0.8 -i 1 -c 150
```

10.10.0.8	[91], 84 bytes, 4.54 ms (88.6 avg, 69% loss)
10.10.0.8	[92], 84 bytes, 66.9 ms (87.8 avg, 68% loss)
10.10.0.8	[93], 84 bytes, 34.8 ms (86.1 avg, 68% loss)
10.10.0.8	[94], 84 bytes, 187 ms (89.3 avg, 67% loss)
10.10.0.8	[95], 84 bytes, 13.6 ms (87.0 avg, 66% loss)
10.10.0.8	[96], 84 bytes, 15.8 ms (84.8 avg, 65% loss)
10.10.0.8	[97], 84 bytes, 5.05 ms (82.5 avg, 65% loss)
10.10.0.8	[98], 84 bytes, 5.58 ms (80.3 avg, 64% loss)
10.10.0.8	[99], 84 bytes, 10.9 ms (78.3 avg, 64% loss)
10.10.0.8	[100], 84 bytes, 9.76 ms (76.5 avg, 63% loss)
10.10.0.8	[101], 84 bytes, 17.8 ms (74.9 avg, 62% loss)
10.10.0.8	[102], 84 bytes, 10.4 ms (73.3 avg, 62% loss)
10.10.0.8	[103], 84 bytes, 7.18 ms (71.6 avg, 61% loss)
10.10.0.8	[104], 84 bytes, 9.05 ms (70.1 avg, 60% loss)
10.10.0.8	[105], 84 bytes, 11.1 ms (68.7 avg, 60% loss)
10.10.0.8	[106], 84 bytes, 37.8 ms (68.0 avg, 59% loss)
10.10.0.8	[107], 84 bytes, 133 ms (69.5 avg, 59% loss)
10.10.0.8	[108], 84 bytes, 4.75 ms (68.0 avg, 58% loss)
10.10.0.8	[109], 84 bytes, 8.13 ms (66.7 avg, 58% loss)
10.10.0.8	[110], 84 bytes, 3.52 ms (65.4 avg, 57% loss)
10.10.0.8	[111], 84 bytes, 28.0 ms (64.6 avg, 57% loss)

```
10.10.0.8 : xmt/rcv/%loss = 105/41/60%, min/avg/max = 1.63/179/3011
root@OpenWrt:~#
```



```
10.10.0.8 : xmt/rev/%loss = 84/32/61%, min/avg/max = 1.56/18.0/80.9
root@OpenWrt:~#
```

[illegible]

```
10.10.0.8 : xmt/rcv/%loss = 75/28/62%, min/avg/max = 2.27/28.8/128
root@OpenWrt:~#
```

```
root@OpenWrt:~# fping 10.10.0.8 -i 1 -c 150
```





```

ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
10.10.0.8 : [52], 84 bytes, 12.3 ms (106 avg, 77% loss)
10.10.0.8 : [53], 84 bytes, 11.6 ms (99.2 avg, 75% loss)
10.10.0.8 : [54], 84 bytes, 4.30 ms (92.4 avg, 74% loss)
10.10.0.8 : [55], 84 bytes, 26.7 ms (88.1 avg, 73% loss)
10.10.0.8 : [56], 84 bytes, 7.37 ms (83.0 avg, 71% loss)
10.10.0.8 : [57], 84 bytes, 4.84 ms (78.4 avg, 70% loss)
10.10.0.8 : [58], 84 bytes, 5.16 ms (74.3 avg, 69% loss)
10.10.0.8 : [59], 84 bytes, 42.1 ms (72.6 avg, 68% loss)
10.10.0.8 : [60], 84 bytes, 26.9 ms (70.4 avg, 67% loss)
10.10.0.8 : [61], 84 bytes, 50.7 ms (69.4 avg, 66% loss)
10.10.0.8 : [62], 84 bytes, 8.15 ms (66.6 avg, 65% loss)
10.10.0.8 : [63], 84 bytes, 7.01 ms (64.0 avg, 64% loss)
10.10.0.8 : [64], 84 bytes, 85.7 ms (64.9 avg, 63% loss)
10.10.0.8 : [65], 84 bytes, 5.02 ms (62.5 avg, 62% loss)
10.10.0.8 : [66], 84 bytes, 7.50 ms (60.4 avg, 61% loss)
10.10.0.8 : [67], 84 bytes, 7.82 ms (58.5 avg, 60% loss)
10.10.0.8 : [68], 84 bytes, 3.57 ms (56.5 avg, 59% loss)
10.10.0.8 : [69], 84 bytes, 4.70 ms (54.7 avg, 58% loss)
10.10.0.8 : [70], 84 bytes, 3.53 ms (53.0 avg, 57% loss)
10.10.0.8 : [71], 84 bytes, 6.48 ms (51.5 avg, 56% loss)
10.10.0.8 : [72], 84 bytes, 8.22 ms (50.2 avg, 56% loss)
10.10.0.8 : [73], 84 bytes, 5.18 ms (48.8 avg, 55% loss)

```

```

10.10.0.8 : xmt/rcv/%loss = 74/33/55%, min/avg/max = 1.58/48.8/1070
root@OpenWrt:~#

```

## RETARDO: 42 SEGUNDOS

```

root@OpenWrt:~# fping 10.10.0.8 -i 1 -c 120
10.10.0.8 : [0], 84 bytes, 5.89 ms (5.89 avg, 0% loss)
10.10.0.8 : [1], 84 bytes, 3.74 ms (4.81 avg, 0% loss)
10.10.0.8 : [2], 84 bytes, 2.28 ms (3.97 avg, 0% loss)
10.10.0.8 : [3], 84 bytes, 1.67 ms (3.39 avg, 0% loss)
10.10.0.8 : [4], 84 bytes, 1.87 ms (3.09 avg, 0% loss)
10.10.0.8 : [5], 84 bytes, 1.56 ms (2.83 avg, 0% loss)
10.10.0.8 : [6], 84 bytes, 7.29 ms (3.47 avg, 0% loss)
10.10.0.8 : [7], 84 bytes, 1.60 ms (3.23 avg, 0% loss)
10.10.0.8 : [8], 84 bytes, 12.1 ms (4.23 avg, 0% loss)
10.10.0.8 : [9], 84 bytes, 1.71 ms (3.97 avg, 0% loss)
10.10.0.8 : [10], 84 bytes, 1.57 ms (3.75 avg, 0% loss)
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8

```

[illegible]

[illegible]



```
10.10.0.8 : xmt/rcv/%loss = 69/23/66%, min/avg/max = 1.52/48.8/1001
root@OpenWrt:~#
```

[illegible]



```
10.10.0.8 : xmt/rcv/%loss = 62/23/62%, min/avg/max = 1.54/9.38/44.6
root@OpenWrt:~#
```

[illegible]

```

ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
10.10.0.8 : [51], 84 bytes, 23.0 ms (5.37 avg, 76% loss)
10.10.0.8 : [52], 84 bytes, 9.23 ms (5.67 avg, 75% loss)
10.10.0.8 : [53], 84 bytes, 125 ms (14.2 avg, 74% loss)
10.10.0.8 : [54], 84 bytes, 3.65 ms (13.5 avg, 72% loss)
10.10.0.8 : [55], 84 bytes, 15.2 ms (13.6 avg, 71% loss)
10.10.0.8 : [56], 84 bytes, 63.6 ms (16.5 avg, 70% loss)
10.10.0.8 : [57], 84 bytes, 86.0 ms (20.4 avg, 68% loss)
10.10.0.8 : [58], 84 bytes, 9.00 ms (19.8 avg, 67% loss)
10.10.0.8 : [59], 84 bytes, 3.61 ms (19.0 avg, 66% loss)
10.10.0.8 : [60], 84 bytes, 12.6 ms (18.7 avg, 65% loss)
10.10.0.8 : [61], 84 bytes, 3.46 ms (18.0 avg, 64% loss)

```

```

10.10.0.8 : xmt/rcv/%loss = 62/22/64%, min/avg/max = 1.53/18.0/125
root@OpenWrt:~#

```

## RETARDO: 41 SEGUNDOS

```

root@OpenWrt:~# fping 10.10.0.8 -i 1 -c 120
10.10.0.8 : [0], 84 bytes, 1.90 ms (1.90 avg, 0% loss)
10.10.0.8 : [1], 84 bytes, 4.73 ms (3.31 avg, 0% loss)
10.10.0.8 : [2], 84 bytes, 3.55 ms (3.39 avg, 0% loss)
10.10.0.8 : [3], 84 bytes, 1.51 ms (2.92 avg, 0% loss)
10.10.0.8 : [4], 84 bytes, 2.15 ms (2.76 avg, 0% loss)
10.10.0.8 : [5], 84 bytes, 5.38 ms (3.20 avg, 0% loss)
10.10.0.8 : [6], 84 bytes, 10.2 ms (4.20 avg, 0% loss)
10.10.0.8 : [7], 84 bytes, 2.03 ms (3.93 avg, 0% loss)
10.10.0.8 : [8], 84 bytes, 1.56 ms (3.66 avg, 0% loss)
10.10.0.8 : [9], 84 bytes, 1.56 ms (3.45 avg, 0% loss)
10.10.0.8 : [10], 84 bytes, 1.75 ms (3.30 avg, 0% loss)
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
10.10.0.8 : [27], 84 bytes, 32.5 ms (5.73 avg, 57% loss)
10.10.0.8 : [28], 84 bytes, 6.20 ms (5.77 avg, 55% loss)
10.10.0.8 : [29], 84 bytes, 4.72 ms (5.69 avg, 53% loss)
10.10.0.8 : [30], 84 bytes, 8.54 ms (5.88 avg, 51% loss)
10.10.0.8 : [31], 84 bytes, 3.97 ms (5.76 avg, 50% loss)
10.10.0.8 : [32], 84 bytes, 5.41 ms (5.74 avg, 48% loss)
10.10.0.8 : [33], 84 bytes, 14.3 ms (6.22 avg, 47% loss)
10.10.0.8 : [34], 84 bytes, 60.1 ms (9.05 avg, 45% loss)
10.10.0.8 : [35], 84 bytes, 365 ms (26.8 avg, 44% loss)
10.10.0.8 : [36], 84 bytes, 29.1 ms (26.9 avg, 43% loss)
10.10.0.8 : [37], 84 bytes, 8.69 ms (26.1 avg, 42% loss)
10.10.0.8 : [38], 84 bytes, 27.5 ms (26.1 avg, 41% loss)
10.10.0.8 : [39], 84 bytes, 3.48 ms (25.2 avg, 40% loss)
10.10.0.8 : [40], 84 bytes, 18.5 ms (24.9 avg, 39% loss)
10.10.0.8 : [41], 84 bytes, 4.30 ms (24.1 avg, 38% loss)
10.10.0.8 : [42], 84 bytes, 19.6 ms (24.0 avg, 37% loss)

```

```

10.10.0.8 : xmt/rcv/%loss = 43/27/37%, min/avg/max = 1.51/24.0/365
root@OpenWrt:~#

```

## RETARDO: 17 SEGUNDOS

```

root@OpenWrt:~# fping 10.10.0.8 -i 1 -c 120
10.10.0.8 : [0], 84 bytes, 6.33 ms (6.33 avg, 0% loss)
10.10.0.8 : [1], 84 bytes, 4.97 ms (5.65 avg, 0% loss)
10.10.0.8 : [2], 84 bytes, 1.99 ms (4.43 avg, 0% loss)
10.10.0.8 : [3], 84 bytes, 2.83 ms (4.03 avg, 0% loss)
10.10.0.8 : [4], 84 bytes, 2.55 ms (3.73 avg, 0% loss)
10.10.0.8 : [5], 84 bytes, 1.52 ms (3.36 avg, 0% loss)

```







```
10.10.0.8 : xmt/rcv/%loss = 58/22/62%, min/avg/max = 1.57/14.2/138
root@OpenWrt:~#
```

[illegible]



[illegible]

## RETARDO: 32 SEGUNDOS

[illegible]

```

ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
10.10.0.8 : [45], 84 bytes, 20.1 ms (6.85 avg, 73% loss)
10.10.0.8 : [46], 84 bytes, 7.07 ms (6.87 avg, 72% loss)
10.10.0.8 : [47], 84 bytes, 11.7 ms (7.22 avg, 70% loss)
10.10.0.8 : [48], 84 bytes, 6.82 ms (7.19 avg, 69% loss)
10.10.0.8 : [49], 84 bytes, 5.14 ms (7.06 avg, 68% loss)
10.10.0.8 : [50], 84 bytes, 11.1 ms (7.30 avg, 66% loss)
10.10.0.8 : [51], 84 bytes, 12.6 ms (7.60 avg, 65% loss)
10.10.0.8 : [52], 84 bytes, 4.23 ms (7.42 avg, 64% loss)
10.10.0.8 : [53], 84 bytes, 28.4 ms (8.47 avg, 62% loss)

10.10.0.8 : xmt/rcv/%loss = 54/20/62%, min/avg/max = 1.55/8.47/28.4
root@OpenWrt:~#

```

## RETARDO: 35 SEGUNDOS

```

root@OpenWrt:~# fping 10.10.0.8 -i 1 -c 120
10.10.0.8 : [0], 84 bytes, 5.29 ms (5.29 avg, 0% loss)
10.10.0.8 : [1], 84 bytes, 4.27 ms (4.78 avg, 0% loss)
10.10.0.8 : [2], 84 bytes, 1.98 ms (3.84 avg, 0% loss)
10.10.0.8 : [3], 84 bytes, 6.85 ms (4.59 avg, 0% loss)
10.10.0.8 : [4], 84 bytes, 4.68 ms (4.61 avg, 0% loss)
10.10.0.8 : [5], 84 bytes, 4.66 ms (4.62 avg, 0% loss)
10.10.0.8 : [6], 84 bytes, 1.55 ms (4.18 avg, 0% loss)
10.10.0.8 : [7], 84 bytes, 2.31 ms (3.94 avg, 0% loss)
10.10.0.8 : [8], 84 bytes, 15.5 ms (5.23 avg, 0% loss)
10.10.0.8 : [9], 84 bytes, 3.71 ms (5.08 avg, 0% loss)
10.10.0.8 : [10], 84 bytes, 2.26 ms (4.82 avg, 0% loss)
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
ICMP Port Unreachable from 10.10.0.8 for ICMP Echo sent to 10.10.0.8
10.10.0.8 : [27], 84 bytes, 11.7 ms (5.40 avg, 57% loss)
10.10.0.8 : [28], 84 bytes, 66.1 ms (10.0 avg, 55% loss)
10.10.0.8 : [29], 84 bytes, 5.97 ms (9.78 avg, 53% loss)
10.10.0.8 : [30], 84 bytes, 10.7 ms (9.85 avg, 51% loss)
10.10.0.8 : [31], 84 bytes, 6.69 ms (9.65 avg, 50% loss)
10.10.0.8 : [32], 84 bytes, 4.81 ms (9.36 avg, 48% loss)
10.10.0.8 : [33], 84 bytes, 5.42 ms (9.15 avg, 47% loss)
10.10.0.8 : [34], 84 bytes, 12.6 ms (9.33 avg, 45% loss)
10.10.0.8 : [35], 84 bytes, 30.6 ms (10.4 avg, 44% loss)
10.10.0.8 : [36], 84 bytes, 4.30 ms (10.1 avg, 43% loss)
10.10.0.8 : [37], 84 bytes, 22.6 ms (10.6 avg, 42% loss)

```













```
10.10.0.8 : xmt/rcv/%loss = 60/17/71%, min/avg/max = 1.59/7.02/42.8
root@OpenWrt:~#
```

[illegible]

```
10.10.0.8 : [54], 84 bytes, 4.34 ms (5.78 avg, 70% loss)
10.10.0.8 : [55], 84 bytes, 4.21 ms (5.68 avg, 69% loss)
10.10.0.8 : [56], 84 bytes, 4.13 ms (5.60 avg, 68% loss)
10.10.0.8 : [57], 84 bytes, 12.1 ms (5.94 avg, 67% loss)
10.10.0.8 : [58], 84 bytes, 3.49 ms (5.82 avg, 66% loss)
```

```
10.10.0.8 : xmt/rcv/%loss = 59/20/66%, min/avg/max = 1.55/5.82/23.9
root@OpenWrt:~#
```

**RETARDO: 40 SEGUNDOS**

## A2.7. MEDIDAS DEL RETARDO CUANDO CAE UN NODO

Los resultados que presentamos a continuación son los que corresponden cuando un nodo vecino cae y hay que recalcular las rutas que utilizaban este nodo como mejor vecino.

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 37.2 ms (37.2 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 11.9 ms (24.6 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 30.7 ms (26.6 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 5.07 ms (21.2 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 97.0 ms (36.4 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 19.1 ms (33.5 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 45.2 ms (35.2 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 7.83 ms (31.8 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 8.54 ms (29.2 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 12.8 ms (27.5 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 4.64 ms (25.4 avg, 0% loss)
10.10.0.1 : [24], 84 bytes, 12.1 ms (24.3 avg, 52% loss)
10.10.0.1 : [25], 84 bytes, 13.0 ms (23.5 avg, 50% loss)
10.10.0.1 : [26], 84 bytes, 5.13 ms (22.2 avg, 48% loss)
10.10.0.1 : [27], 84 bytes, 27.2 ms (22.5 avg, 46% loss)
10.10.0.1 : [28], 84 bytes, 7.17 ms (21.5 avg, 44% loss)
10.10.0.1 : [29], 84 bytes, 21.0 ms (21.5 avg, 43% loss)
root@OpenWrt:~#
```

### RETARDO: 14 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 10.2 ms (10.2 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 69.3 ms (39.7 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 12.7 ms (30.7 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 31.3 ms (30.9 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 19.1 ms (28.5 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 7.07 ms (24.9 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 6.63 ms (22.3 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 6.04 ms (20.3 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 12.1 ms (19.4 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 25.7 ms (20.0 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 37.5 ms (21.6 avg, 0% loss)
10.10.0.1 : [26], 84 bytes, 26.6 ms (22.0 avg, 55% loss)
10.10.0.1 : [27], 84 bytes, 7.12 ms (20.8 avg, 53% loss)
10.10.0.1 : [28], 84 bytes, 4.46 ms (19.7 avg, 51% loss)
10.10.0.1 : [29], 84 bytes, 14.0 ms (19.3 avg, 50% loss)
root@OpenWrt:~#
```

### RETARDO: 16 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 24.1 ms (24.1 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 7.39 ms (15.7 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 15.3 ms (15.6 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 17.0 ms (15.9 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 11.4 ms (15.0 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 8.70 ms (14.0 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 10.3 ms (13.4 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 23.8 ms (14.7 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 30.4 ms (16.5 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 20.2 ms (16.8 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 6.20 ms (15.9 avg, 0% loss)
10.10.0.1 : [25], 84 bytes, 10.2 ms (15.4 avg, 53% loss)
10.10.0.1 : [26], 84 bytes, 9.27 ms (14.9 avg, 51% loss)
10.10.0.1 : [27], 84 bytes, 9.74 ms (14.5 avg, 50% loss)
10.10.0.1 : [28], 84 bytes, 9.51 ms (14.2 avg, 48% loss)
10.10.0.1 : [29], 84 bytes, 273 ms (30.4 avg, 46% loss)
```

```
10.10.0.1 : xmt/rcv/%loss = 30/16/46%, min/avg/max = 6.20/30.4/273
root@OpenWrt:~#
```

## RETARDO: 15 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 86.5 ms (86.5 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 38.9 ms (62.7 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 185 ms (103 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 5.62 ms (79.1 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 13.5 ms (66.0 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 5.73 ms (56.0 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 31.4 ms (52.4 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 6.34 ms (46.7 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 6.20 ms (42.2 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 5.79 ms (38.5 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 18.0 ms (36.7 avg, 0% loss)
10.10.0.1 : [24], 84 bytes, 20.3 ms (35.3 avg, 52% loss)
10.10.0.1 : [25], 84 bytes, 17.8 ms (33.9 avg, 50% loss)
10.10.0.1 : [26], 84 bytes, 226 ms (47.7 avg, 48% loss)
10.10.0.1 : [27], 84 bytes, 13.7 ms (45.4 avg, 46% loss)
10.10.0.1 : [28], 84 bytes, 9.54 ms (43.2 avg, 44% loss)
10.10.0.1 : [29], 84 bytes, 18.0 ms (41.7 avg, 43% loss)
```

```
10.10.0.1 : xmt/rcv/%loss = 30/17/43%, min/avg/max = 5.62/41.7/226
root@OpenWrt:~#
```

## RETARDO: 14 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 15.1 ms (15.1 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 22.5 ms (18.8 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 35.0 ms (24.2 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 78.6 ms (37.8 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 17.8 ms (33.8 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 45.9 ms (35.8 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 4.94 ms (31.4 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 8.40 ms (28.5 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 14.7 ms (27.0 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 15.6 ms (25.8 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 10.0 ms (24.4 avg, 0% loss)
10.10.0.1 : [23], 84 bytes, 11.1 ms (23.3 avg, 50% loss)
10.10.0.1 : [24], 84 bytes, 11.0 ms (22.3 avg, 48% loss)
10.10.0.1 : [25], 84 bytes, 9.29 ms (21.4 avg, 46% loss)
10.10.0.1 : [26], 84 bytes, 24.4 ms (21.6 avg, 44% loss)
10.10.0.1 : [27], 84 bytes, 13.8 ms (21.1 avg, 42% loss)
10.10.0.1 : [28], 84 bytes, 55.4 ms (23.1 avg, 41% loss)
10.10.0.1 : [29], 84 bytes, 13.6 ms (22.6 avg, 40% loss)
```

```
10.10.0.1 : xmt/rcv/%loss = 30/18/40%, min/avg/max = 4.94/22.6/78.6
root@OpenWrt:~#
```

## RETARDO: 13 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 11.0 ms (11.0 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 21.9 ms (16.4 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 6.00 ms (12.9 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 6.08 ms (11.2 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 12.8 ms (11.5 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 6.33 ms (10.7 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 9.06 ms (10.4 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 15.3 ms (11.0 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 12.7 ms (11.2 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 19.5 ms (12.0 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 15.2 ms (12.3 avg, 0% loss)
10.10.0.1 : [26], 84 bytes, 195 ms (27.6 avg, 55% loss)
10.10.0.1 : [27], 84 bytes, 10.6 ms (26.3 avg, 53% loss)
ICMP Host Unreachable from 10.10.0.10 for ICMP Echo sent to 10.10.0.1
10.10.0.1 : [28], 84 bytes, 12.0 ms (25.3 avg, 51% loss)
10.10.0.1 : [29], 84 bytes, 11.4 ms (24.4 avg, 50% loss)
```

```
10.10.0.1 : xmt/rcv/%loss = 30/15/50%, min/avg/max = 6.00/24.4/195
root@OpenWrt:~#
```

## RETARDO: 16 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 1002 ms (1002 avg, 50% loss)
10.10.0.1 : [1], 84 bytes, 6.73 ms (504 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 8.98 ms (339 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 25.5 ms (260 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 19.0 ms (212 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 13.7 ms (179 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 33.8 ms (158 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 15.4 ms (140 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 16.0 ms (126 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 6.44 ms (114 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 205 ms (123 avg, 0% loss)
10.10.0.1 : [22], 84 bytes, 7.59 ms (113 avg, 47% loss)
10.10.0.1 : [23], 84 bytes, 5.98 ms (105 avg, 45% loss)
10.10.0.1 : [24], 84 bytes, 13.0 ms (98.5 avg, 44% loss)
10.10.0.1 : [25], 84 bytes, 8.40 ms (92.5 avg, 42% loss)
10.10.0.1 : [26], 84 bytes, 7.73 ms (87.2 avg, 40% loss)
10.10.0.1 : [27], 84 bytes, 19.4 ms (83.2 avg, 39% loss)
10.10.0.1 : [28], 84 bytes, 20.9 ms (79.8 avg, 37% loss)
10.10.0.1 : [29], 84 bytes, 16.1 ms (76.4 avg, 36% loss)

10.10.0.1 : xmt/rcv/%loss = 30/19/36%, min/avg/max = 5.98/76.4/1002
root@OpenWrt:~#
```

## RETARDO: 12 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 14.5 ms (14.5 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 21.5 ms (18.0 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 109 ms (48.3 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 8.53 ms (38.4 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 6.72 ms (32.0 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 4.71 ms (27.5 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 34.3 ms (28.5 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 16.2 ms (26.9 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 31.4 ms (27.4 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 10.9 ms (25.8 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 24.6 ms (25.7 avg, 0% loss)
10.10.0.1 : [22], 84 bytes, 13.5 ms (24.6 avg, 47% loss)
10.10.0.1 : [23], 84 bytes, 66.0 ms (27.8 avg, 45% loss)
10.10.0.1 : [24], 84 bytes, 7.48 ms (26.4 avg, 44% loss)
10.10.0.1 : [25], 84 bytes, 114 ms (32.3 avg, 42% loss)
10.10.0.1 : [26], 84 bytes, 59.0 ms (33.9 avg, 40% loss)
10.10.0.1 : [27], 84 bytes, 252 ms (46.8 avg, 39% loss)
10.10.0.1 : [28], 84 bytes, 13.3 ms (44.9 avg, 37% loss)
10.10.0.1 : [29], 84 bytes, 20.2 ms (43.6 avg, 36% loss)

10.10.0.1 : xmt/rcv/%loss = 30/19/36%, min/avg/max = 4.71/43.6/252
root@OpenWrt:~#
```

## RETARDO: 12 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 8.44 ms (8.44 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 50.8 ms (29.6 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 13.7 ms (24.3 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 28.8 ms (25.4 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 9.49 ms (22.2 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 8.23 ms (19.9 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 8.95 ms (18.3 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 8.39 ms (17.1 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 35.6 ms (19.1 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 12.9 ms (18.5 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 22.2 ms (18.8 avg, 0% loss)
10.10.0.1 : [24], 84 bytes, 100 ms (25.6 avg, 52% loss)
10.10.0.1 : [25], 84 bytes, 41.6 ms (26.8 avg, 50% loss)
10.10.0.1 : [26], 84 bytes, 6.83 ms (25.4 avg, 48% loss)
10.10.0.1 : [27], 84 bytes, 6.48 ms (24.1 avg, 46% loss)
10.10.0.1 : [28], 84 bytes, 31.0 ms (24.6 avg, 44% loss)
10.10.0.1 : [29], 84 bytes, 4.81 ms (23.4 avg, 43% loss)
```



```
10.10.0.1 : xmt/rcv/%loss = 30/17/43%, min/avg/max = 4.81/23.4/100
root@OpenWrt:~#
```

## RETARDO: 14 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 30.2 ms (30.2 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 12.4 ms (21.3 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 228 ms (90.4 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 20.1 ms (72.8 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 15.3 ms (61.3 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 17.5 ms (54.0 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 10.1 ms (47.7 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 182 ms (64.6 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 26.2 ms (60.3 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 11.7 ms (55.4 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 42.2 ms (54.2 avg, 0% loss)
10.10.0.1 : [19], 84 bytes, 23.3 ms (51.7 avg, 40% loss)
10.10.0.1 : [20], 84 bytes, 28.1 ms (49.9 avg, 38% loss)
10.10.0.1 : [21], 84 bytes, 135 ms (55.9 avg, 36% loss)
10.10.0.1 : [22], 84 bytes, 12.7 ms (53.1 avg, 34% loss)
10.10.0.1 : [23], 84 bytes, 191 ms (61.7 avg, 33% loss)
10.10.0.1 : [24], 84 bytes, 18.5 ms (59.1 avg, 32% loss)
10.10.0.1 : [25], 84 bytes, 10.1 ms (56.4 avg, 30% loss)
10.10.0.1 : [26], 84 bytes, 22.1 ms (54.6 avg, 29% loss)
10.10.0.1 : [27], 84 bytes, 10.3 ms (52.4 avg, 28% loss)
10.10.0.1 : [28], 84 bytes, 8.00 ms (50.3 avg, 27% loss)
10.10.0.1 : [29], 84 bytes, 6.31 ms (48.3 avg, 26% loss)

10.10.0.1 : xmt/rcv/%loss = 30/22/26%, min/avg/max = 6.31/48.3/228
root@OpenWrt:~#
```

## RETARDO: 9 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 9.30 ms (9.30 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 9.21 ms (9.25 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 14.7 ms (11.0 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 12.5 ms (11.4 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 12.8 ms (11.7 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 18.7 ms (12.9 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 20.2 ms (13.9 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 8.55 ms (13.2 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 8.96 ms (12.7 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 22.0 ms (13.7 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 22.7 ms (14.5 avg, 0% loss)
10.10.0.1 : [18], 84 bytes, 10.5 ms (14.1 avg, 36% loss)
10.10.0.1 : [19], 84 bytes, 4.95 ms (13.4 avg, 35% loss)
10.10.0.1 : [20], 84 bytes, 12.9 ms (13.4 avg, 33% loss)
10.10.0.1 : [21], 84 bytes, 194 ms (25.5 avg, 31% loss)
10.10.0.1 : [22], 84 bytes, 49.9 ms (27.0 avg, 30% loss)
10.10.0.1 : [23], 84 bytes, 7.33 ms (25.9 avg, 29% loss)
10.10.0.1 : [24], 84 bytes, 20.6 ms (25.6 avg, 28% loss)
10.10.0.1 : [25], 84 bytes, 127 ms (30.9 avg, 26% loss)
10.10.0.1 : [26], 84 bytes, 9.90 ms (29.9 avg, 25% loss)
10.10.0.1 : [27], 84 bytes, 276 ms (41.6 avg, 25% loss)
10.10.0.1 : [28], 84 bytes, 14.0 ms (40.3 avg, 24% loss)
10.10.0.1 : [29], 84 bytes, 14.3 ms (39.2 avg, 23% loss)

10.10.0.1 : xmt/rcv/%loss = 30/23/23%, min/avg/max = 4.95/39.2/276
root@OpenWrt:~#
```

## RETARDO: 18 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 2035 ms (2035 avg, 66% loss)
10.10.0.1 : [1], 84 bytes, 1035 ms (1535 avg, 33% loss)
10.10.0.1 : [2], 84 bytes, 25.9 ms (1032 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 18.1 ms (778 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 21.5 ms (627 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 18.0 ms (525 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 7.01 ms (451 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 21.5 ms (397 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 9.71 ms (354 avg, 0% loss)
```



```

10.10.0.1 : [9], 84 bytes, 25.9 ms (321 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 4.97 ms (293 avg, 0% loss)
10.10.0.1 : [22], 84 bytes, 12.2 ms (269 avg, 47% loss)
10.10.0.1 : [23], 84 bytes, 12.8 ms (249 avg, 45% loss)
10.10.0.1 : [24], 84 bytes, 21.4 ms (233 avg, 44% loss)
10.10.0.1 : [25], 84 bytes, 35.5 ms (220 avg, 42% loss)
10.10.0.1 : [26], 84 bytes, 12.1 ms (207 avg, 40% loss)
10.10.0.1 : [27], 84 bytes, 8.98 ms (195 avg, 39% loss)
10.10.0.1 : [28], 84 bytes, 16.6 ms (185 avg, 37% loss)
10.10.0.1 : [29], 84 bytes, 14.4 ms (176 avg, 36% loss)

10.10.0.1 : xmt/rcv/%loss = 30/19/36%, min/avg/max = 4.97/176/2035
root@OpenWrt:~#

```

## RETARDO: 12 SEGUNDOS

```

root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 11.3 ms (11.3 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 14.5 ms (12.9 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 28.2 ms (18.0 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 21.5 ms (18.9 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 6.23 ms (16.3 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 277 ms (59.9 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 38.7 ms (56.9 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 16.7 ms (51.9 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 9.23 ms (47.1 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 13.3 ms (43.7 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 451 ms (80.8 avg, 0% loss)
10.10.0.1 : [27], 84 bytes, 14.4 ms (75.3 avg, 57% loss)
10.10.0.1 : [28], 84 bytes, 7.46 ms (70.1 avg, 55% loss)
10.10.0.1 : [29], 84 bytes, 27.2 ms (67.0 avg, 53% loss)
root@OpenWrt:~#

```

## RETARDO: 14 SEGUNDOS

```

root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 23.0 ms (23.0 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 9.45 ms (16.2 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 15.7 ms (16.1 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 24.4 ms (18.1 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 17.1 ms (17.9 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 16.6 ms (17.7 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 28.3 ms (19.2 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 8.46 ms (17.9 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 10.9 ms (17.1 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 11.5 ms (16.5 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 15.0 ms (16.4 avg, 0% loss)
10.10.0.1 : [19], 84 bytes, 15.7 ms (16.3 avg, 40% loss)
10.10.0.1 : [20], 84 bytes, 13.2 ms (16.1 avg, 38% loss)
10.10.0.1 : [21], 84 bytes, 15.3 ms (16.0 avg, 36% loss)
10.10.0.1 : [22], 84 bytes, 151 ms (25.0 avg, 34% loss)
10.10.0.1 : [23], 84 bytes, 4.29 ms (23.7 avg, 33% loss)
10.10.0.1 : [24], 84 bytes, 14.4 ms (23.2 avg, 32% loss)
10.10.0.1 : [25], 84 bytes, 27.8 ms (23.4 avg, 30% loss)
10.10.0.1 : [26], 84 bytes, 19.3 ms (23.2 avg, 29% loss)
10.10.0.1 : [27], 84 bytes, 21.8 ms (23.1 avg, 28% loss)
10.10.0.1 : [28], 84 bytes, 11.1 ms (22.6 avg, 27% loss)
10.10.0.1 : [29], 84 bytes, 7.89 ms (21.9 avg, 26% loss)

```

```

10.10.0.1 : xmt/rcv/%loss = 30/22/26%, min/avg/max = 4.29/21.9/151
root@OpenWrt:~#

```

## RETARDO: 19 SEGUNDOS

```

root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 35.7 ms (35.7 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 4.57 ms (20.1 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 15.2 ms (18.5 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 25.4 ms (20.2 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 12.5 ms (18.6 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 19.7 ms (18.8 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 6.03 ms (17.0 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 5.13 ms (15.5 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 11.2 ms (15.0 avg, 0% loss)

```

```

10.10.0.1 : [9], 84 bytes, 4.47 ms (14.0 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 8.77 ms (13.5 avg, 0% loss)
10.10.0.1 : [24], 84 bytes, 16.3 ms (13.7 avg, 52% loss)
10.10.0.1 : [25], 84 bytes, 6.03 ms (13.1 avg, 50% loss)
10.10.0.1 : [26], 84 bytes, 6.53 ms (12.7 avg, 48% loss)
10.10.0.1 : [27], 84 bytes, 15.2 ms (12.8 avg, 46% loss)
10.10.0.1 : [28], 84 bytes, 9.15 ms (12.6 avg, 44% loss)
10.10.0.1 : [29], 84 bytes, 6.86 ms (12.2 avg, 43% loss)

10.10.0.1 : xmt/rcv/%loss = 30/17/43%, min/avg/max = 4.47/12.2/35.7
root@OpenWrt:~#

```

## RETARDO: 14 SEGUNDOS

```

root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 11.2 ms (11.2 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 4.50 ms (7.88 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 22.3 ms (12.6 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 11.3 ms (12.3 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 176 ms (45.1 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 12.1 ms (39.6 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 61.2 ms (42.7 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 10.4 ms (38.7 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 4.93 ms (34.9 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 7.64 ms (32.2 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 86.7 ms (37.1 avg, 0% loss)
10.10.0.1 : [22], 84 bytes, 17.8 ms (35.5 avg, 47% loss)
10.10.0.1 : [23], 84 bytes, 10.4 ms (33.6 avg, 45% loss)
10.10.0.1 : [24], 84 bytes, 216 ms (46.7 avg, 44% loss)
10.10.0.1 : [25], 84 bytes, 25.4 ms (45.3 avg, 42% loss)
10.10.0.1 : [26], 84 bytes, 34.1 ms (44.6 avg, 40% loss)
10.10.0.1 : [27], 84 bytes, 9.59 ms (42.5 avg, 39% loss)
10.10.0.1 : [28], 84 bytes, 13.5 ms (40.9 avg, 37% loss)
10.10.0.1 : [29], 84 bytes, 42.3 ms (41.0 avg, 36% loss)

10.10.0.1 : xmt/rcv/%loss = 30/19/36%, min/avg/max = 4.50/41.0/216
root@OpenWrt:~#

```

## RETARDO: 12 SEGUNDOS

```

root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 317 ms (317 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 424 ms (371 avg, 33% loss)
10.10.0.1 : [3], 84 bytes, 71.3 ms (271 avg, 25% loss)
10.10.0.1 : [4], 84 bytes, 21.6 ms (208 avg, 20% loss)
10.10.0.1 : [5], 84 bytes, 8.52 ms (168 avg, 16% loss)
10.10.0.1 : [6], 84 bytes, 34.8 ms (146 avg, 14% loss)
10.10.0.1 : [7], 84 bytes, 6.06 ms (126 avg, 12% loss)
10.10.0.1 : [8], 84 bytes, 9.68 ms (111 avg, 11% loss)
10.10.0.1 : [9], 84 bytes, 5.07 ms (99.9 avg, 10% loss)
10.10.0.1 : [10], 84 bytes, 17.5 ms (91.7 avg, 9% loss)
10.10.0.1 : [26], 84 bytes, 26.3 ms (85.8 avg, 59% loss)
10.10.0.1 : [27], 84 bytes, 30.4 ms (81.1 avg, 57% loss)
10.10.0.1 : [28], 84 bytes, 6.96 ms (75.4 avg, 55% loss)
10.10.0.1 : [29], 84 bytes, 10.0 ms (70.8 avg, 53% loss)

10.10.0.1 : xmt/rcv/%loss = 30/14/53%, min/avg/max = 5.07/70.8/424
root@OpenWrt:~#

```

## RETARDO: 16 SEGUNDOS

```

root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
10.10.0.1 : [0], 84 bytes, 5.69 ms (5.69 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 31.8 ms (18.7 avg, 33% loss)
10.10.0.1 : [3], 84 bytes, 35.8 ms (24.4 avg, 25% loss)
10.10.0.1 : [4], 84 bytes, 78.7 ms (38.0 avg, 20% loss)
10.10.0.1 : [5], 84 bytes, 10.3 ms (32.5 avg, 16% loss)
10.10.0.1 : [6], 84 bytes, 19.2 ms (30.3 avg, 14% loss)
10.10.0.1 : [7], 84 bytes, 136 ms (45.4 avg, 12% loss)
10.10.0.1 : [8], 84 bytes, 26.4 ms (43.0 avg, 11% loss)
10.10.0.1 : [9], 84 bytes, 22.0 ms (40.7 avg, 10% loss)
10.10.0.1 : [10], 84 bytes, 14.7 ms (38.1 avg, 9% loss)
10.10.0.1 : [25], 84 bytes, 7.00 ms (35.2 avg, 57% loss)
10.10.0.1 : [26], 84 bytes, 27.7 ms (34.6 avg, 55% loss)

```

```
10.10.0.1 : [27], 84 bytes, 8.48 ms (32.6 avg, 53% loss)
10.10.0.1 : [28], 84 bytes, 20.0 ms (31.7 avg, 51% loss)
10.10.0.1 : [29], 84 bytes, 24.3 ms (31.2 avg, 50% loss)
```

```
10.10.0.1 : xmt/rcv/%loss = 30/15/50%, min/avg/max = 5.69/31.2/136
```

```
root@OpenWrt:~#
```

## RETARDO: 15 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
```

```
10.10.0.1 : [0], 84 bytes, 14.7 ms (14.7 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 13.6 ms (14.1 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 17.5 ms (15.3 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 35.8 ms (20.4 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 371 ms (90.5 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 7.76 ms (76.7 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 83.1 ms (77.6 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 180 ms (90.5 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 9.00 ms (81.4 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 19.2 ms (75.2 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 23.3 ms (70.5 avg, 0% loss)
10.10.0.1 : [29], 84 bytes, 11.1 ms (65.5 avg, 60% loss)
```

```
10.10.0.1 : xmt/rcv/%loss = 30/12/60%, min/avg/max = 7.76/65.5/371
```

```
root@OpenWrt:~#
```

## RETARDO: 19 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 30
```

```
10.10.0.1 : [0], 84 bytes, 21.1 ms (21.1 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 6.15 ms (13.6 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 8.72 ms (11.9 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 57.7 ms (23.4 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 5.01 ms (19.7 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 13.9 ms (18.7 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 13.3 ms (18.0 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 27.3 ms (19.1 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 80.5 ms (25.9 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 61.3 ms (29.5 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 113 ms (37.1 avg, 0% loss)
10.10.0.1 : [25], 84 bytes, 74.0 ms (40.2 avg, 53% loss)
10.10.0.1 : [26], 84 bytes, 10.0 ms (37.9 avg, 51% loss)
10.10.0.1 : [27], 84 bytes, 12.5 ms (36.1 avg, 50% loss)
10.10.0.1 : [28], 84 bytes, 63.2 ms (37.9 avg, 48% loss)
10.10.0.1 : [29], 84 bytes, 13.5 ms (36.3 avg, 46% loss)
```

```
10.10.0.1 : xmt/rcv/%loss = 30/16/46%, min/avg/max = 5.01/36.3/113
```

```
root@OpenWrt:~#
```

## RETARDO: 15 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 300
```

```
10.10.0.1 : [0], 84 bytes, 11.5 ms (11.5 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 16.7 ms (14.1 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 8.27 ms (12.2 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 12.6 ms (12.3 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 12.3 ms (12.3 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 61.4 ms (20.5 avg, 14% loss)
10.10.0.1 : [7], 84 bytes, 62.3 ms (26.4 avg, 12% loss)
10.10.0.1 : [8], 84 bytes, 48.4 ms (29.2 avg, 11% loss)
10.10.0.1 : [9], 84 bytes, 20.0 ms (28.2 avg, 10% loss)
10.10.0.1 : [10], 84 bytes, 14.8 ms (26.8 avg, 9% loss)
10.10.0.1 : [23], 84 bytes, 84.6 ms (32.1 avg, 54% loss)
10.10.0.1 : [24], 84 bytes, 14.8 ms (30.6 avg, 52% loss)
10.10.0.1 : [25], 84 bytes, 4.38 ms (28.6 avg, 50% loss)
10.10.0.1 : [26], 84 bytes, 10.2 ms (27.3 avg, 48% loss)
10.10.0.1 : [27], 84 bytes, 40.7 ms (28.2 avg, 46% loss)
10.10.0.1 : [28], 84 bytes, 40.4 ms (29.0 avg, 44% loss)
10.10.0.1 : [29], 84 bytes, 12.0 ms (28.0 avg, 43% loss)
10.10.0.1 : [30], 84 bytes, 17.7 ms (27.4 avg, 41% loss)
10.10.0.1 : [31], 84 bytes, 51.8 ms (28.7 avg, 40% loss)
10.10.0.1 : [32], 84 bytes, 76.5 ms (31.1 avg, 39% loss)
```

```
10.10.0.1 : xmt/rcv/%loss = 33/20/39%, min/avg/max = 4.38/31.1/84.6
```

root@OpenWrt:~#

## RETARDO: 13 SEGUNDOS

root@OpenWrt:~# **fping 10.10.0.1 -i 1 -c 300**

```
10.10.0.1 : [0], 84 bytes, 10.3 ms (10.3 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 234 ms (122 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 69.3 ms (104 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 380 ms (173 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 29.0 ms (144 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 97.1 ms (136 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 8.64 ms (118 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 10.3 ms (104 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 11.6 ms (94.5 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 53.3 ms (90.4 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 8.97 ms (83.0 avg, 0% loss)
10.10.0.1 : [24], 84 bytes, 766 ms (139 avg, 52% loss)
10.10.0.1 : [25], 84 bytes, 36.7 ms (132 avg, 50% loss)
10.10.0.1 : [26], 84 bytes, 8.22 ms (123 avg, 48% loss)
10.10.0.1 : [27], 84 bytes, 80.7 ms (120 avg, 46% loss)
10.10.0.1 : [28], 84 bytes, 43.6 ms (115 avg, 44% loss)
10.10.0.1 : [29], 84 bytes, 12.9 ms (109 avg, 43% loss)
```

10.10.0.1 : xmt/rcv/%loss = 30/17/43%, min/avg/max = 8.22/109/766

root@OpenWrt:~#

## RETARDO: 14 SEGUNDOS

root@OpenWrt:~# **fping 10.10.0.1 -i 1 -c 300**

```
10.10.0.1 : [0], 84 bytes, 14.5 ms (14.5 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 13.2 ms (13.9 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 11.6 ms (13.1 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 11.8 ms (12.8 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 13.0 ms (12.8 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 7.96 ms (12.0 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 6.65 ms (11.2 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 23.3 ms (12.7 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 55.6 ms (17.5 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 13.7 ms (17.1 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 43.7 ms (19.5 avg, 0% loss)
```

```
10.10.0.1 : [27], 84 bytes, 13.2 ms (19.0 avg, 57% loss)
10.10.0.1 : [28], 84 bytes, 6.46 ms (18.0 avg, 55% loss)
10.10.0.1 : [29], 84 bytes, 20.3 ms (18.2 avg, 53% loss)
```

10.10.0.1 : xmt/rcv/%loss = 30/14/53%, min/avg/max = 6.46/18.2/55.6

root@OpenWrt:~#

## RETARDO: 17 SEGUNDOS

root@OpenWrt:~# **fping 10.10.0.1 -i 1 -c 300**

```
10.10.0.1 : [0], 84 bytes, 12.4 ms (12.4 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 40.6 ms (26.5 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 21.2 ms (24.7 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 50.7 ms (31.2 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 40.1 ms (33.0 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 25.6 ms (31.8 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 23.7 ms (30.6 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 32.1 ms (30.8 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 109 ms (39.5 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 4.48 ms (36.0 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 26.7 ms (35.2 avg, 0% loss)
10.10.0.1 : [27], 84 bytes, 8.49 ms (33.0 avg, 57% loss)
10.10.0.1 : [28], 84 bytes, 48.0 ms (34.1 avg, 55% loss)
10.10.0.1 : [29], 84 bytes, 12.9 ms (32.6 avg, 53% loss)
```

10.10.0.1 : xmt/rcv/%loss = 30/14/53%, min/avg/max = 4.48/32.6/109

root@OpenWrt:~#

## RETARDO: 17 SEGUNDOS

root@OpenWrt:~# **fping 10.10.0.1 -i 1 -c 300**

```
10.10.0.1 : [0], 84 bytes, 61.0 ms (61.0 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 13.6 ms (37.3 avg, 0% loss)
```

```
10.10.0.1 : [2], 84 bytes, 18.9 ms (31.2 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 8.88 ms (25.6 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 18.2 ms (24.1 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 78.6 ms (33.2 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 217 ms (59.5 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 287 ms (88.1 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 71.6 ms (86.2 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 88.8 ms (86.5 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 132 ms (90.6 avg, 0% loss)
10.10.0.1 : [21], 84 bytes, 73.5 ms (89.2 avg, 45% loss)
10.10.0.1 : [22], 84 bytes, 87.0 ms (89.0 avg, 43% loss)
10.10.0.1 : [24], 84 bytes, 12.0 ms (83.5 avg, 44% loss)
10.10.0.1 : [25], 84 bytes, 285 ms (97.0 avg, 42% loss)
10.10.0.1 : [26], 84 bytes, 41.2 ms (93.5 avg, 40% loss)
10.10.0.1 : [27], 84 bytes, 13.7 ms (88.8 avg, 39% loss)
10.10.0.1 : [28], 84 bytes, 61.6 ms (87.3 avg, 37% loss)
10.10.0.1 : [29], 84 bytes, 75.7 ms (86.7 avg, 36% loss)
10.10.0.1 : [31], 84 bytes, 883 ms (126 avg, 37% loss)

10.10.0.1 : xmt/rcv/%loss = 33/20/39%, min/avg/max = 8.88/126/883
root@OpenWrt:~#
```

## RETARDO: 11 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 300
10.10.0.1 : [0], 84 bytes, 2023 ms (2023 avg, 66% loss)
10.10.0.1 : [1], 84 bytes, 1019 ms (1521 avg, 33% loss)
10.10.0.1 : [2], 84 bytes, 36.3 ms (1026 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 33.7 ms (778 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 4.69 ms (623 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 146 ms (544 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 23.3 ms (469 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 93.9 ms (422 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 11.5 ms (377 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 50.7 ms (344 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 19.5 ms (314 avg, 0% loss)
10.10.0.1 : [34], 84 bytes, 109 ms (297 avg, 65% loss)
10.10.0.1 : [35], 84 bytes, 43.9 ms (278 avg, 63% loss)
10.10.0.1 : [36], 84 bytes, 48.4 ms (261 avg, 62% loss)
10.10.0.1 : [37], 84 bytes, 80.1 ms (249 avg, 60% loss)
10.10.0.1 : [38], 84 bytes, 54.1 ms (237 avg, 58% loss)
10.10.0.1 : [39], 84 bytes, 89.2 ms (228 avg, 57% loss)

10.10.0.1 : xmt/rcv/%loss = 40/17/57%, min/avg/max = 4.69/228/2023
root@OpenWrt:~#
```

## RETARDO: 24 SEGUNDOS

```
root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 300
10.10.0.1 : [0], 84 bytes, 2006 ms (2006 avg, 50% loss)
10.10.0.1 : [1], 84 bytes, 999 ms (1502 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 94.4 ms (1033 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 44.8 ms (786 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 17.7 ms (632 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 72.3 ms (539 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 38.3 ms (467 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 211 ms (435 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 17.7 ms (389 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 33.6 ms (353 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 50.5 ms (326 avg, 0% loss)
10.10.0.1 : [34], 84 bytes, 46.4 ms (302 avg, 65% loss)
10.10.0.1 : [35], 84 bytes, 18.3 ms (280 avg, 63% loss)
10.10.0.1 : [36], 84 bytes, 71.0 ms (265 avg, 62% loss)
10.10.0.1 : [37], 84 bytes, 81.8 ms (253 avg, 60% loss)
10.10.0.1 : [38], 84 bytes, 47.1 ms (240 avg, 58% loss)
10.10.0.1 : [39], 84 bytes, 14.5 ms (227 avg, 57% loss)
10.10.0.1 : [40], 84 bytes, 34.7 ms (216 avg, 56% loss)
10.10.0.1 : [41], 84 bytes, 24.3 ms (206 avg, 54% loss)
10.10.0.1 : [42], 84 bytes, 15.7 ms (197 avg, 53% loss)

10.10.0.1 : xmt/rcv/%loss = 43/20/53%, min/avg/max = 14.5/197/2006
root@OpenWrt:~#
```

## RETARDO: 24 SEGUNDOS

root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 300

```
10.10.0.1 : [0], 84 bytes, 93.4 ms (93.4 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 12.0 ms (52.7 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 19.5 ms (41.6 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 141 ms (66.5 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 13.0 ms (55.8 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 29.0 ms (51.3 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 29.0 ms (48.1 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 15.3 ms (44.0 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 25.4 ms (42.0 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 11.7 ms (38.9 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 11.6 ms (36.5 avg, 0% loss)
10.10.0.1 : [27], 84 bytes, 16.6 ms (34.8 avg, 57% loss)
10.10.0.1 : [28], 84 bytes, 24.7 ms (34.0 avg, 55% loss)
10.10.0.1 : [29], 84 bytes, 32.2 ms (33.9 avg, 53% loss)
10.10.0.1 : [30], 84 bytes, 13.4 ms (32.5 avg, 51% loss)
10.10.0.1 : [31], 84 bytes, 8.02 ms (31.0 avg, 50% loss)
10.10.0.1 : [32], 84 bytes, 6.04 ms (29.5 avg, 48% loss)
```

10.10.0.1 : xmt/rcv/%loss = 33/17/48%, min/avg/max = 6.04/29.5/141

root@OpenWrt:~#

## RETARDO: 17 SEGUNDOS

root@OpenWrt:~# fping 10.10.0.1 -i 1 -c 300

```
10.10.0.1 : [0], 84 bytes, 14.0 ms (14.0 avg, 0% loss)
10.10.0.1 : [1], 84 bytes, 10.9 ms (12.5 avg, 0% loss)
10.10.0.1 : [2], 84 bytes, 10.0 ms (11.6 avg, 0% loss)
10.10.0.1 : [3], 84 bytes, 6.75 ms (10.4 avg, 0% loss)
10.10.0.1 : [4], 84 bytes, 5.29 ms (9.41 avg, 0% loss)
10.10.0.1 : [5], 84 bytes, 15.3 ms (10.4 avg, 0% loss)
10.10.0.1 : [6], 84 bytes, 21.8 ms (12.0 avg, 0% loss)
10.10.0.1 : [7], 84 bytes, 17.2 ms (12.7 avg, 0% loss)
10.10.0.1 : [8], 84 bytes, 25.2 ms (14.0 avg, 0% loss)
10.10.0.1 : [9], 84 bytes, 22.9 ms (14.9 avg, 0% loss)
10.10.0.1 : [10], 84 bytes, 6.58 ms (14.2 avg, 0% loss)
10.10.0.1 : [28], 84 bytes, 19.3 ms (14.6 avg, 58% loss)
10.10.0.1 : [29], 84 bytes, 7.78 ms (14.1 avg, 56% loss)
10.10.0.1 : [30], 84 bytes, 16.7 ms (14.3 avg, 54% loss)
10.10.0.1 : [31], 84 bytes, 39.0 ms (15.9 avg, 53% loss)
10.10.0.1 : [32], 84 bytes, 10.5 ms (15.6 avg, 51% loss)
10.10.0.1 : [33], 84 bytes, 7.52 ms (15.1 avg, 50% loss)
10.10.0.1 : [34], 84 bytes, 90.6 ms (19.3 avg, 48% loss)
```

10.10.0.1 : xmt/rcv/%loss = 35/18/48%, min/avg/max = 5.29/19.3/90.6

root@OpenWrt:~#

## RETARDO: 18 SEGUNDOS

## A2.8. MEDIDAS DEL RETARDO CUANDO INTRODUCIMOS UN NODO NUEVO

Para obtener los resultados cuando introducimos un nuevo nodo nos sucede lo mismo que en la situación de arranque de la red. Es decir, el comando fping no muestra salida por comandos hasta que no se puede acceder al nodo. Por lo que utilizamos el mismo Script “arranque” para calcular el tiempo que tarda en ser accesible un nodo nuevo.

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:56 UTC 2009
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [3], 84 bytes, 1019 ms (1019 avg, 80% loss)
10.10.0.10 : [4], 84 bytes, 24.6 ms (521 avg, 60% loss)
10.10.0.10 : [5], 84 bytes, 22.8 ms (355 avg, 50% loss)
10.10.0.10 : [6], 84 bytes, 6.39 ms (268 avg, 42% loss)
10.10.0.10 : [7], 84 bytes, 5.97 ms (215 avg, 37% loss)
10.10.0.10 : [8], 84 bytes, 15.0 ms (182 avg, 33% loss)
10.10.0.10 : [9], 84 bytes, 17.1 ms (158 avg, 30% loss)

10.10.0.10 : xmt/rcv/%loss = 10/7/30%, min/avg/max = 5.97/158/1019
Fri Feb 6 02:50:06 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

### RETARDO = 8 SEGUNDOS

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:57 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:58 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:59 UTC 2009
10.10.0.10 : [3], 84 bytes, 2042 ms (2042 avg, 83% loss)
10.10.0.10 : [4], 84 bytes, 1032 ms (1537 avg, 66% loss)
10.10.0.10 : [5], 84 bytes, 23.2 ms (1032 avg, 50% loss)
10.10.0.10 : [6], 84 bytes, 37.3 ms (783 avg, 42% loss)
10.10.0.10 : [7], 84 bytes, 6.24 ms (628 avg, 37% loss)
10.10.0.10 : [8], 84 bytes, 16.0 ms (526 avg, 33% loss)
10.10.0.10 : [9], 84 bytes, 33.9 ms (456 avg, 30% loss)

10.10.0.10 : xmt/rcv/%loss = 10/7/30%, min/avg/max = 6.24/456/2042
Fri Feb 6 02:50:09 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

### RETARDO = 8 SEGUNDOS

mbenito@liendre:~/Desktop\$ sh arranque

Fri Feb 6 02:49:57 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:58 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:59 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:00 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:01 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:02 UTC 2009

10.10.0.10 : [0], 84 bytes, 3033 ms (3033 avg, 75% loss)

10.10.0.10 : [1], 84 bytes, 2027 ms (2530 avg, 50% loss)

10.10.0.10 : [2], 84 bytes, 1017 ms (2026 avg, 25% loss)

10.10.0.10 : [3], 84 bytes, 17.6 ms (1524 avg, 0% loss)

10.10.0.10 : [4], 84 bytes, 28.1 ms (1224 avg, 0% loss)

10.10.0.10 : [5], 84 bytes, 17.5 ms (1023 avg, 0% loss)

10.10.0.10 : [6], 84 bytes, 28.9 ms (881 avg, 0% loss)

10.10.0.10 : [7], 84 bytes, 7.09 ms (772 avg, 0% loss)

10.10.0.10 : [8], 84 bytes, 8.29 ms (687 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/9/10%, min/avg/max = 7.09/687/3033

Fri Feb 6 02:50:12 UTC 2009

^Cmbenito@liendre:~/Desktop\$

## RETARDO = 5 SEGUNDOS

mbenito@liendre:~/Desktop\$ sh arranque

Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:56 UTC 2009

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

10.10.0.10 : [0], 84 bytes, 43.6 ms (43.6 avg, 0% loss)

10.10.0.10 : [1], 84 bytes, 20.7 ms (32.1 avg, 0% loss)

10.10.0.10 : [2], 84 bytes, 2064 ms (709 avg, 40% loss)

10.10.0.10 : [3], 84 bytes, 1056 ms (796 avg, 20% loss)

10.10.0.10 : [4], 84 bytes, 47.6 ms (646 avg, 0% loss)

10.10.0.10 : [5], 84 bytes, 17.8 ms (541 avg, 0% loss)

10.10.0.10 : [6], 84 bytes, 11.4 ms (466 avg, 0% loss)

10.10.0.10 : [7], 84 bytes, 11.6 ms (409 avg, 0% loss)

10.10.0.10 : [8], 84 bytes, 11.1 ms (365 avg, 0% loss)

10.10.0.10 : [9], 84 bytes, 7.19 ms (329 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 7.19/329/2064

Fri Feb 6 02:50:06 UTC 2009

^Cmbenito@liendre:~/Desktop\$

## RETARDO = 3 SEGUNDOS

mbenito@liendre:~/Desktop\$ sh arranque

Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:54 UTC 2009



10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:56 UTC 2009

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
10.10.0.10 : [0], 84 bytes, 1030 ms (1030 avg, 50% loss)  
10.10.0.10 : [1], 84 bytes, 121 ms (575 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 1003 ms (718 avg, 25% loss)  
10.10.0.10 : [4], 84 bytes, 18.1 ms (543 avg, 20% loss)  
10.10.0.10 : [5], 84 bytes, 5.65 ms (435 avg, 16% loss)  
10.10.0.10 : [6], 84 bytes, 331 ms (418 avg, 14% loss)  
10.10.0.10 : [7], 84 bytes, 31.8 ms (363 avg, 12% loss)  
10.10.0.10 : [8], 84 bytes, 4.74 ms (318 avg, 11% loss)  
10.10.0.10 : [9], 84 bytes, 25.8 ms (285 avg, 10% loss)

10.10.0.10 : xmt/rcv/%loss = 10/9/10%, min/avg/max = 4.74/285/1030  
Fri Feb 6 02:50:06 UTC 2009

**^Cmbenito@liendre:~/Desktop\$**

## RETARDO = 4 SEGUNDOS

**mbenito@liendre:~/Desktop\$ sh arranque**  
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:57 UTC 2009

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
10.10.0.10 : [0], 84 bytes, 1013 ms (1013 avg, 50% loss)  
10.10.0.10 : [1], 84 bytes, 18.3 ms (516 avg, 0% loss)  
10.10.0.10 : [2], 84 bytes, 20.9 ms (351 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 15.7 ms (267 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 7.96 ms (215 avg, 0% loss)  
10.10.0.10 : [5], 84 bytes, 599 ms (279 avg, 0% loss)  
10.10.0.10 : [7], 84 bytes, 62.6 ms (248 avg, 12% loss)  
10.10.0.10 : [8], 84 bytes, 11.5 ms (218 avg, 11% loss)  
10.10.0.10 : [9], 84 bytes, 19.0 ms (196 avg, 10% loss)

10.10.0.10 : xmt/rcv/%loss = 10/9/10%, min/avg/max = 7.96/196/1013  
Fri Feb 6 02:50:06 UTC 2009

**^Cmbenito@liendre:~/Desktop\$**

## RETARDO = 4 SEGUNDOS

**mbenito@liendre:~/Desktop\$ sh arranque**  
Fri Feb 6 02:50:03 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:04 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:05 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:06 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:07 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:08 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:09 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:10 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:11 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:12 UTC 2009  
[^Cmbenito@liendre:~/Desktop\\$](#)

## RETARDO = 9 SEGUNDOS

[mbenito@liendre:~/Desktop\\$](#) sh arranque  
Fri Feb 6 02:50:45 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:46 UTC 2009  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
10.10.0.10 : [2], 84 bytes, 11.3 ms (11.3 avg, 66% loss)  
10.10.0.10 : [3], 84 bytes, 14.5 ms (12.9 avg, 50% loss)  
10.10.0.10 : [4], 84 bytes, 7.28 ms (11.0 avg, 40% loss)  
10.10.0.10 : [5], 84 bytes, 72.6 ms (26.4 avg, 33% loss)  
10.10.0.10 : [6], 84 bytes, 66.2 ms (34.4 avg, 28% loss)  
10.10.0.10 : [7], 84 bytes, 7.35 ms (29.8 avg, 25% loss)  
10.10.0.10 : [8], 84 bytes, 42.3 ms (31.6 avg, 22% loss)  
10.10.0.10 : [9], 84 bytes, 30.6 ms (31.5 avg, 20% loss)  
  
10.10.0.10 : xmt/rcv/%loss = 10/8/20%, min/avg/max = 7.28/31.5/72.6  
Fri Feb 6 02:50:56 UTC 2009  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
10.10.0.10 : [0], 84 bytes, 27.8 ms (27.8 avg, 0% loss)  
10.10.0.10 : [1], 84 bytes, 11.4 ms (19.6 avg, 0% loss)  
10.10.0.10 : [2], 84 bytes, 24.1 ms (21.1 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 11.1 ms (18.6 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 27.5 ms (20.4 avg, 0% loss)  
10.10.0.10 : [5], 84 bytes, 35.3 ms (22.9 avg, 0% loss)  
10.10.0.10 : [6], 84 bytes, 15.3 ms (21.8 avg, 0% loss)  
10.10.0.10 : [8], 84 bytes, 31.3 ms (23.0 avg, 11% loss)  
10.10.0.10 : [9], 84 bytes, 38.0 ms (24.6 avg, 10% loss)  
  
10.10.0.10 : xmt/rcv/%loss = 10/9/10%, min/avg/max = 11.1/24.6/38.0  
Fri Feb 6 02:51:06 UTC 2009  
10.10.0.10 : [1], 84 bytes, 6.08 ms (6.08 avg, 50% loss)  
10.10.0.10 : [2], 84 bytes, 43.4 ms (24.7 avg, 33% loss)  
10.10.0.10 : [3], 84 bytes, 57.1 ms (35.5 avg, 25% loss)  
10.10.0.10 : [4], 84 bytes, 6.72 ms (28.3 avg, 20% loss)  
10.10.0.10 : [6], 84 bytes, 34.9 ms (29.6 avg, 28% loss)  
10.10.0.10 : [7], 84 bytes, 17.8 ms (27.6 avg, 25% loss)  
10.10.0.10 : [8], 84 bytes, 100 ms (38.0 avg, 22% loss)  
10.10.0.10 : [9], 84 bytes, 10.7 ms (34.6 avg, 20% loss)  
  
10.10.0.10 : xmt/rcv/%loss = 10/8/20%, min/avg/max = 6.08/34.6/100  
Fri Feb 6 02:51:15 UTC 2009  
[^Cmbenito@liendre:~/Desktop\\$](#)

## RETARDO = 4 SEGUNDOS

[mbenito@liendre:~/Desktop\\$](#) sh arranque  
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:55 UTC 2009  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
10.10.0.10 : [2], 84 bytes, 20.4 ms (20.4 avg, 66% loss)  
10.10.0.10 : [3], 84 bytes, 9.12 ms (14.7 avg, 50% loss)  
10.10.0.10 : [4], 84 bytes, 14.2 ms (14.5 avg, 40% loss)  
10.10.0.10 : [5], 84 bytes, 8.89 ms (13.1 avg, 33% loss)  
10.10.0.10 : [6], 84 bytes, 25.2 ms (15.5 avg, 28% loss)  
10.10.0.10 : [7], 84 bytes, 25.5 ms (17.2 avg, 25% loss)  
10.10.0.10 : [8], 84 bytes, 10.0 ms (16.2 avg, 22% loss)  
10.10.0.10 : [9], 84 bytes, 70.2 ms (22.9 avg, 20% loss)

10.10.0.10 : xmt/rcv/%loss = 10/8/20%, min/avg/max = 8.89/22.9/70.2  
Fri Feb 6 02:50:04 UTC 2009  
[^Cmbenito@liendre:~/Desktop\\$](#)

## RETARDO = 5 SEGUNDOS

[mbenito@liendre:~/Desktop\\$](#) sh arranque  
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:57 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:58 UTC 2009  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
10.10.0.10 : [0], 84 bytes, 3047 ms (3047 avg, 75% loss)  
10.10.0.10 : [1], 84 bytes, 2042 ms (2544 avg, 50% loss)  
10.10.0.10 : [2], 84 bytes, 1042 ms (2044 avg, 25% loss)  
10.10.0.10 : [3], 84 bytes, 42.7 ms (1543 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 13.5 ms (1237 avg, 0% loss)  
10.10.0.10 : [5], 84 bytes, 10.6 ms (1033 avg, 0% loss)  
10.10.0.10 : [6], 84 bytes, 11.5 ms (887 avg, 0% loss)  
10.10.0.10 : [7], 84 bytes, 22.7 ms (779 avg, 0% loss)  
10.10.0.10 : [8], 84 bytes, 18.0 ms (694 avg, 0% loss)  
10.10.0.10 : [9], 84 bytes, 14.1 ms (626 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 10.6/626/3047  
Fri Feb 6 02:50:08 UTC 2009  
[^Cmbenito@liendre:~/Desktop\\$](#)

## RETARDO = 4 SEGUNDOS

[mbenito@liendre:~/Desktop\\$](#) sh arranque  
Fri Feb 6 02:49:51 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:56 UTC 2009

```
10.10.0.10 : [0], 84 bytes, 3089 ms (3089 avg, 75% loss)
10.10.0.10 : [1], 84 bytes, 2087 ms (2588 avg, 50% loss)
10.10.0.10 : [2], 84 bytes, 1086 ms (2088 avg, 25% loss)
10.10.0.10 : [3], 84 bytes, 83.1 ms (1586 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 41.7 ms (1277 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 47.6 ms (1072 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 6.97 ms (920 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 55.3 ms (812 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 37.0 ms (726 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 30.4 ms (656 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 6.97/656/3089
Fri Feb 6 02:50:05 UTC 2009
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 5 SEGUNDOS

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:54 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:55 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:56 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:57 UTC 2009
10.10.0.10 : [0], 84 bytes, 3026 ms (3026 avg, 75% loss)
10.10.0.10 : [1], 84 bytes, 2025 ms (2525 avg, 50% loss)
10.10.0.10 : [2], 84 bytes, 1015 ms (2022 avg, 25% loss)
10.10.0.10 : [3], 84 bytes, 15.9 ms (1520 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 25.5 ms (1221 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 22.4 ms (1021 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 46.6 ms (882 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 1002 ms (897 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 38.1 ms (802 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 5.36 ms (722 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 5.36/722/3026
Fri Feb 6 02:50:06 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 3 SEGUNDOS

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:52 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:53 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:54 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:55 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:56 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:57 UTC 2009
10.10.0.10 : [0], 84 bytes, 1033 ms (1033 avg, 50% loss)
10.10.0.10 : [1], 84 bytes, 29.2 ms (531 avg, 0% loss)
10.10.0.10 : [2], 84 bytes, 28.4 ms (363 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 12.1 ms (275 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 15.3 ms (223 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 1047 ms (360 avg, 14% loss)
10.10.0.10 : [6], 84 bytes, 89.2 ms (322 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 47.5 ms (287 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 63.9 ms (262 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 30.7 ms (239 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 12.1/239/1047
```

Fri Feb 6 02:50:06 UTC 2009  
^Cmbenito@liendre:~/Desktop\$

## RETARDO = 5 SEGUNDOS

mбенито@liendre:~/Desktop\$ sh arranque  
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:57 UTC 2009  
10.10.0.10 : [0], 84 bytes, 2024 ms (2024 avg, 66% loss)  
10.10.0.10 : [1], 84 bytes, 1021 ms (1522 avg, 33% loss)  
10.10.0.10 : [2], 84 bytes, 19.2 ms (1021 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 5.15 ms (767 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 30.9 ms (620 avg, 0% loss)  
10.10.0.10 : [5], 84 bytes, 12.3 ms (518 avg, 0% loss)  
10.10.0.10 : [6], 84 bytes, 15.7 ms (447 avg, 0% loss)  
10.10.0.10 : [7], 84 bytes, 1041 ms (521 avg, 11% loss)  
10.10.0.10 : [8], 84 bytes, 55.1 ms (469 avg, 0% loss)  
10.10.0.10 : [9], 84 bytes, 8.61 ms (423 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 5.15/423/2024  
Fri Feb 6 02:50:06 UTC 2009  
^Cmbenito@liendre:~/Desktop\$

## RETARDO = 5 SEGUNDOS

mбенито@liendre:~/Desktop\$ sh arranque  
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:57 UTC 2009  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
10.10.0.10 : [2], 84 bytes, 1057 ms (1057 avg, 75% loss)  
10.10.0.10 : [3], 84 bytes, 56.5 ms (557 avg, 50% loss)  
10.10.0.10 : [4], 84 bytes, 23.4 ms (379 avg, 40% loss)  
10.10.0.10 : [5], 84 bytes, 11.1 ms (287 avg, 33% loss)  
10.10.0.10 : [6], 84 bytes, 16.0 ms (232 avg, 28% loss)  
10.10.0.10 : [7], 84 bytes, 26.5 ms (198 avg, 25% loss)  
10.10.0.10 : [8], 84 bytes, 13.5 ms (172 avg, 22% loss)  
10.10.0.10 : [9], 84 bytes, 7.75 ms (151 avg, 20% loss)

10.10.0.10 : xmt/rcv/%loss = 10/8/20%, min/avg/max = 7.75/151/1057  
Fri Feb 6 02:50:07 UTC 2009  
^Cmbenito@liendre:~/Desktop\$

## RETARDO = 7 SEGUNDOS

mбенито@liendre:~/Desktop\$ sh arranque  
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:57 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:58 UTC 2009  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
10.10.0.10 : [0], 84 bytes, 1077 ms (1077 avg, 50% loss)  
10.10.0.10 : [1], 84 bytes, 74.4 ms (576 avg, 0% loss)  
10.10.0.10 : [2], 84 bytes, 11.5 ms (387 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 11.5 ms (293 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 1029 ms (440 avg, 16% loss)  
10.10.0.10 : [5], 84 bytes, 64.8 ms (378 avg, 0% loss)  
10.10.0.10 : [6], 84 bytes, 6.56 ms (325 avg, 0% loss)  
10.10.0.10 : [7], 84 bytes, 7.52 ms (285 avg, 0% loss)  
10.10.0.10 : [8], 84 bytes, 1023 ms (367 avg, 10% loss)  
10.10.0.10 : [9], 84 bytes, 13.8 ms (332 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 6.56/332/1077  
Fri Feb 6 02:50:07 UTC 2009  
[^Cmbenito@liendre:~/Desktop\\$](#)

## RETARDO = 5 SEGUNDOS

[mbenito@liendre:~/Desktop\\$](#) sh arranque  
Fri Feb 6 02:49:51 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:57 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:58 UTC 2009  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
ICMP Host Unreachable from 10.10.0.3 for ICMP Echo sent to 10.10.0.10  
ICMP Host Unreachable from 10.10.0.3 for ICMP Echo sent to 10.10.0.10  
ICMP Host Unreachable from 10.10.0.3 for ICMP Echo sent to 10.10.0.10  
10.10.0.10 : [0], 84 bytes, 1108 ms (1108 avg, 50% loss)  
10.10.0.10 : [1], 84 bytes, 103 ms (606 avg, 0% loss)  
10.10.0.10 : [2], 84 bytes, 27.5 ms (413 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 41.9 ms (320 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 21.3 ms (260 avg, 0% loss)  
10.10.0.10 : [5], 84 bytes, 10.0 ms (218 avg, 0% loss)  
10.10.0.10 : [9], 84 bytes, 38.3 ms (193 avg, 30% loss)

10.10.0.10 : xmt/rcv/%loss = 10/7/30%, min/avg/max = 10.0/193/1108  
Fri Feb 6 02:50:07 UTC 2009  
10.10.0.10 : [0], 84 bytes, 16.5 ms (16.5 avg, 0% loss)  
10.10.0.10 : [1], 84 bytes, 27.5 ms (22.0 avg, 0% loss)  
10.10.0.10 : [2], 84 bytes, 8.72 ms (17.5 avg, 0% loss)

10.10.0.10 : [3], 84 bytes, 46.6 ms (24.8 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 14.8 ms (22.8 avg, 0% loss)  
10.10.0.10 : [5], 84 bytes, 17.4 ms (21.9 avg, 0% loss)  
10.10.0.10 : [6], 84 bytes, 128 ms (37.2 avg, 0% loss)  
10.10.0.10 : [7], 84 bytes, 20.2 ms (35.0 avg, 0% loss)  
10.10.0.10 : [8], 84 bytes, 54.4 ms (37.2 avg, 0% loss)  
10.10.0.10 : [9], 84 bytes, 9.90 ms (34.5 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 8.72/34.5/128

Fri Feb 6 02:50:17 UTC 2009

^Cmbenito@liendre:~/Desktop\$

## RETARDO = 7 SEGUNDOS

mbenito@liendre:~/Desktop\$ sh arranque

Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:57 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:58 UTC 2009

10.10.0.10 : [0], 84 bytes, 27.6 ms (27.6 avg, 0% loss)

10.10.0.10 : [1], 84 bytes, 10.5 ms (19.1 avg, 0% loss)

10.10.0.10 : [2], 84 bytes, 14.8 ms (17.7 avg, 0% loss)

10.10.0.10 : [3], 84 bytes, 34.0 ms (21.7 avg, 0% loss)

10.10.0.10 : [4], 84 bytes, 33.0 ms (24.0 avg, 0% loss)

10.10.0.10 : [5], 84 bytes, 11.0 ms (21.8 avg, 0% loss)

10.10.0.10 : [6], 84 bytes, 4.92 ms (19.4 avg, 0% loss)

10.10.0.10 : [7], 84 bytes, 33.3 ms (21.1 avg, 0% loss)

10.10.0.10 : [8], 84 bytes, 10.5 ms (19.9 avg, 0% loss)

10.10.0.10 : [9], 84 bytes, 8.89 ms (18.8 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 4.92/18.8/34.0

Fri Feb 6 02:50:07 UTC 2009

^Cmbenito@liendre:~/Desktop\$

## RETARDO = 6 SEGUNDOS

mbenito@liendre:~/Desktop\$ sh arranque

Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:57 UTC 2009

10.10.0.10 : [0], 84 bytes, 32.9 ms (32.9 avg, 0% loss)

10.10.0.10 : [1], 84 bytes, 15.6 ms (24.3 avg, 0% loss)

10.10.0.10 : [2], 84 bytes, 14.5 ms (21.0 avg, 0% loss)

10.10.0.10 : [3], 84 bytes, 11.3 ms (18.6 avg, 0% loss)

10.10.0.10 : [4], 84 bytes, 7.75 ms (16.4 avg, 0% loss)

10.10.0.10 : [5], 84 bytes, 38.8 ms (20.1 avg, 0% loss)

10.10.0.10 : [6], 84 bytes, 6.19 ms (18.1 avg, 0% loss)

```
10.10.0.10 : [7], 84 bytes, 15.6 ms (17.8 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 8.32 ms (16.8 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 7.33 ms (15.8 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 6.19/15.8/38.8
Fri Feb 6 02:50:06 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 5 SEGUNDOS

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:52 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:53 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:54 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:55 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:56 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:57 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:58 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:59 UTC 2009
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [0], 84 bytes, 16.2 ms (16.2 avg, 0% loss)
10.10.0.10 : [1], 84 bytes, 24.8 ms (20.5 avg, 0% loss)
10.10.0.10 : [2], 84 bytes, 27.0 ms (22.7 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 18.1 ms (21.5 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 72.5 ms (31.7 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 30.4 ms (31.5 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 1018 ms (172 avg, 12% loss)
10.10.0.10 : [7], 84 bytes, 9.37 ms (152 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 15.4 ms (136 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 27.6 ms (126 avg, 0% loss)
```

```
10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 9.37/126/1018
Fri Feb 6 02:50:08 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 7 SEGUNDOS

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:54 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:55 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:56 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:57 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:58 UTC 2009
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [0], 84 bytes, 2004 ms (2004 avg, 50% loss)
10.10.0.10 : [1], 84 bytes, 1008 ms (1506 avg, 0% loss)
10.10.0.10 : [2], 84 bytes, 25.1 ms (1012 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 13.1 ms (763 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 16.5 ms (613 avg, 0% loss)
```



```
10.10.0.10 : [5], 84 bytes, 14.1 ms (513 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 11.4 ms (442 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 12.2 ms (388 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 22.7 ms (347 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 14.3 ms (314 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 11.4/314/2004
Fri Feb 6 02:50:08 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 4 SEGUNDOS

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:53 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:54 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:55 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:56 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:57 UTC 2009
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [2], 84 bytes, 21.5 ms (21.5 avg, 66% loss)
10.10.0.10 : [3], 84 bytes, 11.0 ms (16.2 avg, 50% loss)
10.10.0.10 : [4], 84 bytes, 6.38 ms (12.9 avg, 40% loss)
10.10.0.10 : [5], 84 bytes, 15.1 ms (13.5 avg, 33% loss)
10.10.0.10 : [6], 84 bytes, 33.8 ms (17.6 avg, 28% loss)
10.10.0.10 : [7], 84 bytes, 15.3 ms (17.2 avg, 25% loss)
10.10.0.10 : [8], 84 bytes, 6.04 ms (15.6 avg, 22% loss)
10.10.0.10 : [9], 84 bytes, 5.09 ms (14.3 avg, 20% loss)
```

```
10.10.0.10 : xmt/rcv/%loss = 10/8/20%, min/avg/max = 5.09/14.3/33.8
Fri Feb 6 02:50:07 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 7 SEGUNDOS

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:59 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:00 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:50:01 UTC 2009
10.10.0.10 : [0], 84 bytes, 33.1 ms (33.1 avg, 0% loss)
10.10.0.10 : [1], 84 bytes, 9.67 ms (21.4 avg, 0% loss)
10.10.0.10 : [2], 84 bytes, 14.7 ms (19.1 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 6.69 ms (16.0 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 10.2 ms (14.8 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 19.9 ms (15.7 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 14.8 ms (15.5 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 23.3 ms (16.5 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 12.5 ms (16.1 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 13.7 ms (15.8 avg, 0% loss)
```

```
10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 6.69/15.8/33.1
Fri Feb 6 02:50:11 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 2 SEGUNDOS

```
mbenito@liendre:~/Desktop$ sh arranque
```

Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:56 UTC 2009

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

10.10.0.10 : [0], 84 bytes, 2041 ms (2041 avg, 66% loss)

10.10.0.10 : [1], 84 bytes, 1041 ms (1541 avg, 33% loss)

10.10.0.10 : [2], 84 bytes, 32.1 ms (1038 avg, 0% loss)

10.10.0.10 : [3], 84 bytes, 7.04 ms (780 avg, 0% loss)

10.10.0.10 : [4], 84 bytes, 10.3 ms (626 avg, 0% loss)

10.10.0.10 : [5], 84 bytes, 11.8 ms (523 avg, 0% loss)

10.10.0.10 : [6], 84 bytes, 1019 ms (594 avg, 12% loss)

10.10.0.10 : [7], 84 bytes, 19.3 ms (522 avg, 0% loss)

10.10.0.10 : [8], 84 bytes, 16.6 ms (466 avg, 0% loss)

10.10.0.10 : [9], 84 bytes, 8.43 ms (420 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 7.04/420/2041

Fri Feb 6 02:50:06 UTC 2009

[^Cmbenito@liendre:~/Desktop\\$](#)

## RETARDO = 3 SEGUNDOS

[mbenito@liendre:~/Desktop\\$](#) sh arranque

Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:57 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:49:58 UTC 2009

10.10.0.10 : [0], 84 bytes, 18.3 ms (18.3 avg, 0% loss)

10.10.0.10 : [1], 84 bytes, 10.1 ms (14.2 avg, 0% loss)

10.10.0.10 : [2], 84 bytes, 136 ms (54.8 avg, 0% loss)

10.10.0.10 : [3], 84 bytes, 9.89 ms (43.6 avg, 0% loss)

10.10.0.10 : [4], 84 bytes, 30.6 ms (41.0 avg, 0% loss)

10.10.0.10 : [5], 84 bytes, 11.0 ms (36.0 avg, 0% loss)

10.10.0.10 : [6], 84 bytes, 6.62 ms (31.8 avg, 0% loss)

10.10.0.10 : [7], 84 bytes, 6.52 ms (28.6 avg, 0% loss)

10.10.0.10 : [8], 84 bytes, 17.2 ms (27.4 avg, 0% loss)

10.10.0.10 : [9], 84 bytes, 21.3 ms (26.8 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 6.52/26.8/136

Fri Feb 6 02:50:07 UTC 2009

ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10

[^Cmbenito@liendre:~/Desktop\\$](#)

## RETARDO = 4 SEGUNDOS

[mbenito@liendre:~/Desktop\\$](#) sh arranque

Fri Feb 6 02:50:04 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:05 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:06 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%

Fri Feb 6 02:50:07 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:08 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:50:09 UTC 2009  
10.10.0.10 : [0], 84 bytes, 2057 ms (2057 avg, 66% loss)  
10.10.0.10 : [1], 84 bytes, 1071 ms (1564 avg, 33% loss)  
10.10.0.10 : [2], 84 bytes, 63.0 ms (1063 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 22.5 ms (803 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 16.8 ms (646 avg, 0% loss)  
10.10.0.10 : [5], 84 bytes, 10.7 ms (540 avg, 0% loss)  
10.10.0.10 : [6], 84 bytes, 20.9 ms (466 avg, 0% loss)  
10.10.0.10 : [7], 84 bytes, 11.9 ms (409 avg, 0% loss)  
10.10.0.10 : [8], 84 bytes, 17.2 ms (365 avg, 0% loss)  
10.10.0.10 : [9], 84 bytes, 67.5 ms (335 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 10.7/335/2057

Fri Feb 6 02:50:18 UTC 2009  
**^Cmbenito@liendre:~/Desktop\$**

## RETARDO = 5 SEGUNDOS

**mbenito@liendre:~/Desktop\$ sh arranque**

Fri Feb 6 02:49:52 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:56 UTC 2009  
10.10.0.10 : [0], 84 bytes, 32.7 ms (32.7 avg, 0% loss)  
10.10.0.10 : [1], 84 bytes, 15.1 ms (23.9 avg, 0% loss)  
10.10.0.10 : [2], 84 bytes, 37.2 ms (28.3 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 8.42 ms (23.3 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 7.14 ms (20.1 avg, 0% loss)  
10.10.0.10 : [5], 84 bytes, 72.7 ms (28.8 avg, 0% loss)  
10.10.0.10 : [6], 84 bytes, 111 ms (40.7 avg, 0% loss)  
10.10.0.10 : [7], 84 bytes, 7.44 ms (36.5 avg, 0% loss)  
10.10.0.10 : [8], 84 bytes, 10.6 ms (33.6 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/9/10%, min/avg/max = 7.14/33.6/111

Fri Feb 6 02:50:06 UTC 2009  
**^Cmbenito@liendre:~/Desktop\$**

## RETARDO = 4 SEGUNDOS

**mbenito@liendre:~/Desktop\$ sh arranque**

Fri Feb 6 02:49:53 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:54 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:55 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:56 UTC 2009

10.10.0.10 : xmt/rcv/%loss = 0/0/0%  
Fri Feb 6 02:49:57 UTC 2009  
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10  
10.10.0.10 : [0], 84 bytes, 1028 ms (1028 avg, 50% loss)  
10.10.0.10 : [1], 84 bytes, 22.1 ms (525 avg, 0% loss)  
10.10.0.10 : [2], 84 bytes, 14.7 ms (355 avg, 0% loss)  
10.10.0.10 : [3], 84 bytes, 14.1 ms (269 avg, 0% loss)  
10.10.0.10 : [4], 84 bytes, 5.76 ms (217 avg, 0% loss)  
10.10.0.10 : [5], 84 bytes, 9.04 ms (182 avg, 0% loss)  
10.10.0.10 : [6], 84 bytes, 8.30 ms (157 avg, 0% loss)

```
10.10.0.10 : [7], 84 bytes, 8.97 ms (138 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 1026 ms (237 avg, 10% loss)
10.10.0.10 : [9], 84 bytes, 29.4 ms (216 avg, 0% loss)

10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 5.76/216/1028
Fri Feb 6 02:50:06 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 4 SEGUNDOS

```
mbenito@liendre:~/Desktop$ sh arranque
Fri Feb 6 02:49:53 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:54 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:55 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:56 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:57 UTC 2009
```

```
10.10.0.10 : xmt/rcv/%loss = 0/0/0%
Fri Feb 6 02:49:58 UTC 2009
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
ICMP Redirect from 10.10.0.2 for ICMP Echo sent to 10.10.0.10
10.10.0.10 : [1], 84 bytes, 25.5 ms (25.5 avg, 50% loss)
10.10.0.10 : [2], 84 bytes, 10.3 ms (17.9 avg, 33% loss)
10.10.0.10 : [3], 84 bytes, 2014 ms (683 avg, 40% loss)
10.10.0.10 : [4], 84 bytes, 1004 ms (763 avg, 20% loss)
10.10.0.10 : [5], 84 bytes, 8.66 ms (612 avg, 16% loss)
10.10.0.10 : [6], 84 bytes, 82.7 ms (524 avg, 14% loss)
10.10.0.10 : [7], 84 bytes, 28.4 ms (453 avg, 12% loss)
10.10.0.10 : [8], 84 bytes, 6.83 ms (397 avg, 11% loss)
10.10.0.10 : [9], 84 bytes, 19.0 ms (355 avg, 10% loss)
```

```
10.10.0.10 : xmt/rcv/%loss = 10/9/10%, min/avg/max = 6.83/355/2014
Fri Feb 6 02:50:07 UTC 2009
10.10.0.10 : [0], 84 bytes, 5.97 ms (5.97 avg, 0% loss)
10.10.0.10 : [1], 84 bytes, 13.9 ms (9.98 avg, 0% loss)
10.10.0.10 : [2], 84 bytes, 5.08 ms (8.34 avg, 0% loss)
10.10.0.10 : [3], 84 bytes, 12.7 ms (9.44 avg, 0% loss)
10.10.0.10 : [4], 84 bytes, 17.5 ms (11.0 avg, 0% loss)
10.10.0.10 : [5], 84 bytes, 19.3 ms (12.4 avg, 0% loss)
10.10.0.10 : [6], 84 bytes, 4.76 ms (11.3 avg, 0% loss)
10.10.0.10 : [7], 84 bytes, 33.1 ms (14.0 avg, 0% loss)
10.10.0.10 : [8], 84 bytes, 6.10 ms (13.1 avg, 0% loss)
10.10.0.10 : [9], 84 bytes, 9.33 ms (12.8 avg, 0% loss)
```

```
10.10.0.10 : xmt/rcv/%loss = 10/10/0%, min/avg/max = 4.76/12.8/33.1
Fri Feb 6 02:50:16 UTC 2009
^Cmbenito@liendre:~/Desktop$
```

## RETARDO = 6 SEGUNDOS

## A2.9. CAPTURA DE TRÁFICO DENTRO DE LA RED MALLADA

Vamos a mostrar el resultado de 2 capturas de tráfico para ver el uso de los puertos 4305 y 4306. Primero vamos a mostrar el resultado de la captura de tráfico en el puerto 4306 en una red mallada inalámbrica de 9 routers dispuestos como se muestra en la Figura 4.9. Y en la misma situación una captura de tráfico sin especificar el puerto.

### A2.9.1. CAPTURA DE TRÁFICO EN EL PUERTO 4306

En esta situación hacemos la captura de tráfico en la gateway 10.10.0.1 y especificamos que el tráfico que queremos capturar es el que tiene como origen o destino el puerto 4306. Para conseguir esto ejecutamos en el nodo 10.10.0.1 en el momento que el nodo cliente 10.10.0.10 accede a Internet el siguiente comando:

```
tcpdump -i wl0 port 4306 > dentro
```

El fichero “dentro” es el que contiene el resultado de la captura, y es el que mostramos a continuación. Las direcciones IPs utilizadas son:

**10.10.0.1** → GATEWAY → **0a 0a 00 01**

**10.10.0.10** → CLIENTE → **0a 0a 00 0a**

169.254.0.7 → gate0 **10.10.0.10** → **a9 fe 00 07**

163.117.139.253 → DNS → **a3 75 8b fd**

209.85.229.104 → google.es → **d1 55 e5 68**

```
root@OpenWrt:~# cat dentro
03:00:31.798298 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 56
  0x0000: 4500 0054 0000 4000 3d11 297b 0a0a 000a E..T..@.=.){....
  0x0010: 0a0a 0001 10d2 10d2 0040 7585 0145 0000 .....@u..E..
  0x0020: 37ec f740 0040 1174 46a9 fe00 07a3 758b 7..@..@.tF.....u.
  0x0030: fd08 0000 3500 2358 3f00 0201 0000 0100 ....5.#X?.....
  0x0040: 0000 0000 0006 676f 6f67 6c65 0265 7300 .....google.es.
  0x0050: 001c ..
03:00:31.800605 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 116
  0x0000: 4500 0090 0000 4000 4011 263f 0a0a 0001 E.....@.@.&?....
  0x0010: 0a0a 000a 10d2 10d2 007c b10d 0145 0000 .....|..E..
  0x0020: 7300 0040 003d 1164 02a3 758b fda9 fe00 s..@.=.d.u.....
  0x0030: 0700 3508 0000 5f91 5d00 0281 8000 0100 ..5..._].....
  0x0040: 0000 0100 0006 676f 6f67 6c65 0265 7300 .....google.es.
  0x0050: 001c ..
03:00:31.807619 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 116
  0x0000: 4500 0090 0000 4000 3f11 273f 0a0a 0001 E.....@.?."?....
  0x0010: 0a0a 000a 10d2 10d2 007c b10d 0145 0000 .....|..E..
  0x0020: 7300 0040 003d 1164 02a3 758b fda9 fe00 s..@.=.d.u.....
  0x0030: 0700 3508 0000 5f91 5d00 0281 8000 0100 ..5..._].....
  0x0040: 0000 0100 0006 676f 6f67 6c65 0265 7300 .....google.es.
  0x0050: 001c ..
03:00:31.811331 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 116
  0x0000: 4500 0090 0000 4000 3e11 283f 0a0a 0001 E.....@.>.(?....
  0x0010: 0a0a 000a 10d2 10d2 007c b10d 0145 0000 .....|..E..
  0x0020: 7300 0040 003d 1164 02a3 758b fda9 fe00 s..@.=.d.u.....
  0x0030: 0700 3508 0000 5f91 5d00 0281 8000 0100 ..5..._].....
  0x0040: 0000 0100 0006 676f 6f67 6c65 0265 7300 .....google.es.
  0x0050: 001c ..
03:00:31.815207 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 116
  0x0000: 4500 0090 0000 4000 3d11 293f 0a0a 0001 E.....@.=.)?....
  0x0010: 0a0a 000a 10d2 10d2 007c b10d 0145 0000 .....|..E..
  0x0020: 7300 0040 003d 1164 02a3 758b fda9 fe00 s..@.=.d.u.....
  0x0030: 0700 3508 0000 5f91 5d00 0281 8000 0100 ..5..._].....
```

```

0x0040: 0000 0100 0006 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 001c
03:00:31.824046 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 56
0x0000: 4500 0054 0000 4000 3d11 297b 0a0a 000a E..T..@.=){...
0x0010: 0a0a 0001 10d2 10d2 0040 7585 0145 0000 .....@u.E..
0x0020: 37ec fa40 0040 1174 43a9 fe00 07a3 758b 7..@..tC....u.
0x0030: fd08 0000 3500 2358 3e00 0301 0000 0100 ...5.#X>.....
0x0040: 0000 0000 0006 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 001c
03:00:31.825882 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 116
0x0000: 4500 0090 0000 4000 4011 263f 0a0a 0001 E.....@.@.&?...
0x0010: 0a0a 000a 10d2 10d2 007c b10d 0145 0000 .....|.E..
0x0020: 7300 0040 003d 1164 02a3 758b fda9 fe00 s..@.=d.u....
0x0030: 0700 3508 0000 5f91 5c00 0381 8000 0100 ..5..._\......
0x0040: 0000 0100 0006 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 001c
03:00:31.827598 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 116
0x0000: 4500 0090 0000 4000 3f11 273f 0a0a 0001 E.....@.@.?....
0x0010: 0a0a 000a 10d2 10d2 007c b10d 0145 0000 .....|.E..
0x0020: 7300 0040 003d 1164 02a3 758b fda9 fe00 s..@.=d.u....
0x0030: 0700 3508 0000 5f91 5c00 0381 8000 0100 ..5..._\......
0x0040: 0000 0100 0006 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 001c
03:00:31.865490 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 56
0x0000: 4500 0054 0000 4000 4011 267b 0a0a 000a E..T..@.@.&{...
0x0010: 0a0a 0001 10d2 10d2 0040 7585 0145 0000 .....@u.E..
0x0020: 37ec fe40 0040 1174 3fa9 fe00 07a3 758b 7..@..t?....u.
0x0030: fd08 0000 3500 2358 3d00 0401 0000 0100 ...5.#X=.....
0x0040: 0000 0000 0006 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 001c
03:00:31.883998 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 56
0x0000: 4500 0054 0000 4000 3d11 297b 0a0a 000a E..T..@.=){...
0x0010: 0a0a 0001 10d2 10d2 0040 7585 0145 0000 .....@u.E..
0x0020: 37ec fe40 0040 1174 3fa9 fe00 07a3 758b 7..@..t?....u.
0x0030: fd08 0000 3500 2358 3d00 0401 0000 0100 ...5.#X=.....
0x0040: 0000 0000 0006 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 001c
03:00:31.913130 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 116
0x0000: 4500 0090 0000 4000 4011 263f 0a0a 0001 E.....@.@.&?...
0x0010: 0a0a 000a 10d2 10d2 007c b10d 0145 0000 .....|.E..
0x0020: 7300 0040 003d 1164 02a3 758b fda9 fe00 s..@.=d.u....
0x0030: 0700 3508 0000 5f91 5b00 0481 8000 0100 ..5...[. ....
0x0040: 0000 0100 0006 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 001c
03:00:31.923799 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 116
0x0000: 4500 0090 0000 4000 3f11 273f 0a0a 0001 E.....@.@.?....
0x0010: 0a0a 000a 10d2 10d2 007c b10d 0145 0000 .....|.E..
0x0020: 7300 0040 003d 1164 02a3 758b fda9 fe00 s..@.=d.u....
0x0030: 0700 3508 0000 5f91 5b00 0481 8000 0100 ..5...[. ....
0x0040: 0000 0100 0006 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 001c
03:00:31.924214 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 116
0x0000: 4500 0090 0000 4000 3e11 283f 0a0a 0001 E.....@.>.(?...
0x0010: 0a0a 000a 10d2 10d2 007c b10d 0145 0000 .....|.E..
0x0020: 7300 0040 003d 1164 02a3 758b fda9 fe00 s..@.=d.u....
0x0030: 0700 3508 0000 5f91 5b00 0481 8000 0100 ..5...[. ....
0x0040: 0000 0100 0006 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 001c
03:00:31.924644 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 116
0x0000: 4500 0090 0000 4000 3d11 293f 0a0a 0001 E.....@.=.)?...
0x0010: 0a0a 000a 10d2 10d2 007c b10d 0145 0000 .....|.E..
0x0020: 7300 0040 003d 1164 02a3 758b fda9 fe00 s..@.=d.u....
0x0030: 0700 3508 0000 5f91 5b00 0481 8000 0100 ..5...[. ....
0x0040: 0000 0100 0006 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 001c
03:00:31.927184 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 56
0x0000: 4500 0054 0000 4000 4011 267b 0a0a 000a E..T..@.@.&{...
0x0010: 0a0a 0001 10d2 10d2 0040 7585 0145 0000 .....@u.E..
0x0020: 37ed 0440 0040 1174 39a9 fe00 07a3 758b 7..@..t9....u.
0x0030: fd08 0000 3500 2373 3c00 0501 0000 0100 ...5.#s<.....
0x0040: 0000 0000 0006 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 0001
03:00:31.927580 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 56
0x0000: 4500 0054 0000 4000 3f11 277b 0a0a 000a E..T..@.@.?.'{...
0x0010: 0a0a 0001 10d2 10d2 0040 7585 0145 0000 .....@u.E..
0x0020: 37ed 0440 0040 1174 39a9 fe00 07a3 758b 7..@..t9....u.

```

```

0x0030: fd08 0000 3500 2373 3c00 0501 0000 0100 ....5.#s<.....
0x0040: 0000 0000 0006 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 0001
..
03:00:31.940141 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 56
0x0000: 4500 0054 0000 4000 3d11 297b 0a0a 000a E..T..@.=){...
0x0010: 0a0a 0001 10d2 10d2 0040 7585 0145 0000 .....@u..E..
0x0020: 37ed 0440 0040 1174 39a9 fe00 07a3 758b 7..@..@.t9....u.
0x0030: fd08 0000 3500 2373 3c00 0501 0000 0100 ....5.#s<.....
0x0040: 0000 0000 0006 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 0001
..
03:00:31.966574 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 250
0x0000: 4500 0116 0000 4000 4011 25b9 0a0a 0001 E.....@..@.%.....
0x0010: 0a0a 000a 10d2 10d2 0102 3602 0145 0000 .....6..E..
0x0020: f900 0040 003d 1163 7ca3 758b fda9 fe00 ...@.=.c|.u....
0x0030: 0700 3508 0000 e516 fb00 0581 8000 0100 ..5.....
0x0040: 0300 0400 0406 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 0001
..
03:00:31.995147 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 250
0x0000: 4500 0116 0000 4000 3f11 26b9 0a0a 0001 E.....@.?.&.....
0x0010: 0a0a 000a 10d2 10d2 0102 3602 0145 0000 .....6..E..
0x0020: f900 0040 003d 1163 7ca3 758b fda9 fe00 ...@.=.c|.u....
0x0030: 0700 3508 0000 e516 fb00 0581 8000 0100 ..5.....
0x0040: 0300 0400 0406 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 0001
..
03:00:31.996640 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 250
0x0000: 4500 0116 0000 4000 3e11 27b9 0a0a 0001 E.....@.>.'.....
0x0010: 0a0a 000a 10d2 10d2 0102 3602 0145 0000 .....6..E..
0x0020: f900 0040 003d 1163 7ca3 758b fda9 fe00 ...@.=.c|.u....
0x0030: 0700 3508 0000 e516 fb00 0581 8000 0100 ..5.....
0x0040: 0300 0400 0406 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 0001
..
03:00:31.997105 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 250
0x0000: 4500 0116 0000 4000 3d11 28b9 0a0a 0001 E.....@.=.(.....
0x0010: 0a0a 000a 10d2 10d2 0102 3602 0145 0000 .....6..E..
0x0020: f900 0040 003d 1163 7ca3 758b fda9 fe00 ...@.=.c|.u....
0x0030: 0700 3508 0000 e516 fb00 0581 8000 0100 ..5.....
0x0040: 0300 0400 0406 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 0001
..
03:00:32.003269 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3f11 275e 0a0a 000a E..q..@.?.'^.....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@..@.....U.
0x0030: 6808 0020 86d1 0200 0048 43be 3300 0000 h.....HC.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000
..
03:00:32.003656 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3e11 285e 0a0a 000a E..q..@.>.(^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@..@.....U.
0x0030: 6808 0020 86d1 0200 0048 43be 3300 0000 h.....HC.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000
..
03:00:32.004357 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3d11 295e 0a0a 000a E..q..@.=)^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@..@.....U.
0x0030: 6808 0020 86d1 0200 0048 43be 3300 0000 h.....HC.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000
..
03:00:32.071548 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 4011 265e 0a0a 0001 E..q..@..@.&^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 54a0 f200 0032 0186 f3d1 55e5 68a9 fe00 T....2....U.h...
0x0030: 0700 0028 86d1 0200 0048 43be 3300 0000 ...(.....HC.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000
..
03:00:32.075566 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3f11 275e 0a0a 0001 E..q..@.?.'^.....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 54a0 f200 0032 0186 f3d1 55e5 68a9 fe00 T....2....U.h...
0x0030: 0700 0028 86d1 0200 0048 43be 3300 0000 ...(.....HC.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000
..
03:00:32.076014 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3e11 285e 0a0a 0001 E..q..@.>.(^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..

```

```

0x0020: 54a0 f200 0032 0186 f3d1 55e5 68a9 fe00 T...2...U.h...
0x0030: 0700 0028 86d1 0200 0048 43be 3300 0000 ...(.....HC.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:00:32.076453 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3d11 295e 0a0a 0001 E..q..@.=)^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 54a0 f200 0032 0186 f3d1 55e5 68a9 fe00 T...2...U.h...
0x0030: 0700 0028 86d1 0200 0048 43be 3300 0000 ...(.....HC.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:00:32.999395 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 4011 265e 0a0a 000a E..q..@.@.&^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@.@.....U.
0x0030: 6808 002a 4cd1 0200 012f 7ccd 3300 0000 h..*L.../.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:00:33.003775 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3f11 275e 0a0a 000a E..q..@.?.'^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@.@.....U.
0x0030: 6808 002a 4cd1 0200 012f 7ccd 3300 0000 h..*L.../.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:00:33.004739 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3d11 295e 0a0a 000a E..q..@.=)^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@.@.....U.
0x0030: 6808 002a 4cd1 0200 012f 7ccd 3300 0000 h..*L.../.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:00:33.203858 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 4011 265e 0a0a 0001 E..q..@.@.&^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 54a0 f300 0032 0186 f2d1 55e5 68a9 fe00 T...2...U.h...
0x0030: 0700 0032 4cd1 0200 012f 7ccd 3300 0000 ...2L.../.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:00:33.205378 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3f11 275e 0a0a 0001 E..q..@.?.'^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 54a0 f300 0032 0186 f2d1 55e5 68a9 fe00 T...2...U.h...
0x0030: 0700 0032 4cd1 0200 012f 7ccd 3300 0000 ...2L.../.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:00:33.206065 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3e11 285e 0a0a 0001 E..q..@.>(^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 54a0 f300 0032 0186 f2d1 55e5 68a9 fe00 T...2...U.h...
0x0030: 0700 0032 4cd1 0200 012f 7ccd 3300 0000 ...2L.../.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:00:33.206602 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3d11 295e 0a0a 0001 E..q..@.=)^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 54a0 f300 0032 0186 f2d1 55e5 68a9 fe00 T...2...U.h...
0x0030: 0700 0032 4cd1 0200 012f 7ccd 3300 0000 ...2L.../.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:00:33.999123 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 4011 265e 0a0a 000a E..q..@.@.&^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@.@.....U.
0x0030: 6808 00cc 09d1 0200 027e bddc 3300 0000 h.....~.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:00:34.000876 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3e11 285e 0a0a 000a E..q..@.>(^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@.@.....U.
0x0030: 6808 00cc 09d1 0200 027e bddc 3300 0000 h.....~.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:00:34.001433 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3d11 295e 0a0a 000a E..q..@.=)^....

```



```

0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@.@.....U.
0x0030: 6808 00cc 09d1 0200 027e bddc 3300 0000 h.....~.3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:00:34.077224 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 4011 265e 0a0a 0001 E..q..@.@.&^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 54a0 f400 0032 0186 f1d1 55e5 68a9 fe00 T....2....U.h...
0x0030: 0700 00d4 09d1 0200 027e bddc 3300 0000 .....~..3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:00:34.080670 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3f11 275e 0a0a 0001 E..q..@.?.'^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 54a0 f400 0032 0186 f1d1 55e5 68a9 fe00 T....2....U.h...
0x0030: 0700 00d4 09d1 0200 027e bddc 3300 0000 .....~..3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:00:34.080724 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3e11 285e 0a0a 0001 E..q..@.>.(^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 54a0 f400 0032 0186 f1d1 55e5 68a9 fe00 T....2....U.h...
0x0030: 0700 00d4 09d1 0200 027e bddc 3300 0000 .....~..3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:00:34.080765 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3d11 295e 0a0a 0001 E..q..@.=.)^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 54a0 f400 0032 0186 f1d1 55e5 68a9 fe00 T....2....U.h...
0x0030: 0700 00d4 09d1 0200 027e bddc 3300 0000 .....~..3...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..

```

root@OpenWrt:~#

## A2.9.2. CAPTURA DE TRÁFICO SIN ESPECIFICAR EL PUERTO

Esta situación es la que muestra la Figura 4.9 y se realiza una captura de tráfico en el nodo 10.10.0.8 cuando el nodo cliente 10.10.0.10 accede a Internet. Para ello se ejecuta el siguiente comando en el nodo 10.10.0.8:

```
tcpdump -i wl0 -X > Dentro
```

El fichero “Dentro” contiene el resultado de la captura y es el que mostramos a continuación. Las direcciones IPs utilizadas son:

**10.10.255.255 (0a0a ffff)** → Dirección de broadcast.

**10.10.0.10 (0a0a 000a)** → Nodo cliente.

**10.10.0.1 (0a0a 0001)** → Nodo Gateway.

**10.10.0.8 (0a0a 0008)** → Mejor vecino del nodo 10.10.0.10 para alcanzar a 10.10.0.1.

169.254.0.7 (**a9 fe00 07**) → Es la interfaz gate0 del cliente **10.10.0.10**.

169.254.0.5 (**a9 fe00 05**) → Es la interfaz gate0 del nodo **10.10.0.8**.

209.85.229.104 (**d1 55e5 68**) → Es la dirección de Google.es.

163.117.139.253 (**a3 75 8b fd**) → Es el servidor DNS.

```
root@OpenWrt:~# cat Dentro
```

```
03:09:44.550400 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
    0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&.....
    0x0010: 0a0a ffff 10d1 10d1 001a 1dd9 0500 2e00 .....
    0x0020: 0336 10d2 0a0a 0003 0a0a 0004 5100 .....6.....Q.
03:09:44.581271 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
    0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E.....@.@.&.....
    0x0010: 0a0a ffff 10d1 10d1 001a 2ddd 0500 2f00 .....-.../.
    0x0020: 032c 10d2 0a0a 0009 0a0a 0008 4000 .....@.
03:09:44.752893 arp who-has 10.10.0.4 tell 10.10.0.8
    0x0000: 0001 0800 0604 0001 001c 1088 9606 0a0a .....
    0x0010: 0008 0000 0000 0000 0a0a 0004 .....
03:09:44.894894 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
    0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E.....@.@.&.....
    0x0010: 0a0a ffff 10d1 10d1 001a 6bd9 0500 3200 .....k...2.
    0x0020: 0332 10d2 0a0a 0006 0a0a 0006 ff00 .....2.....
03:09:44.925730 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
    0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E.....@.@.&.....
    0x0010: 0a0a ffff 10d1 10d1 001a ed95 0540 3000 .....@0.
    0x0020: 0332 10d2 0a0a 0006 0a0a 0006 7f00 .....2.....
03:09:44.941335 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
    0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E.....@.@.&.....
    0x0010: 0a0a ffff 10d1 10d1 001a ad9c 0540 3100 .....@1.
    0x0020: 0332 10d2 0a0a 0006 0a0a 0006 be00 .....2.....
03:09:44.965488 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
    0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E.....@.@.&.....
    0x0010: 0a0a ffff 10d1 10d1 001a cee1 0500 3000 .....0.
    0x0020: 0332 10d2 0a0a 0006 0a0a 0003 9e00 .....2.....
03:09:44.970376 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
    0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&.....
    0x0010: 0a0a ffff 10d1 10d1 001a 28d4 0500 2f00 .....(.../.
    0x0020: 0332 10d2 0a0a 0006 0a0a 000a 4500 .....2.....E.
03:09:45.019201 arp who-has 10.10.0.8 tell 10.10.0.10
    0x0000: 0001 0800 0604 0001 001c 10a4 378a 0a0a .....7...
    0x0010: 000a 0000 0000 0000 0a0a 0008 .....
03:09:45.019397 arp reply 10.10.0.8 is-at 00:1c:10:88:96:06 (oui Unknown)
    0x0000: 0001 0800 0604 0002 001c 1088 9606 0a0a .....
    0x0010: 0008 001c 10a4 378a 0a0a 000a .....7....
03:09:45.021339 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
    0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E.....@.@.&.....
    0x0010: 0a0a ffff 10d1 10d1 001a 16da 0500 2e00 .....
    0x0020: 0332 10d2 0a0a 0006 0a0a 0008 5800 .....2.....X.
03:09:45.022262 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
```

```

0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a fce2 0500 2f00 ...../..
0x0020: 0332 10d2 0a0a 0006 0a0a 0001 7100   2.....q.
03:09:45.027599 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 56
0x0000: 4500 0054 0000 4000 4011 267b 0a0a 000a E..T..@.@.&{....
0x0010: 0a0a 0001 10d2 10d2 0040 7585 0145 0000 .....@u..E..
0x0020: 37e2 e040 0040 117e 5da9 fe00 07a3 758b 7..@.@.~]....u.
0x0030: fd08 0000 3500 2358 3f00 0201 0000 0100 ....5.#X?.....
0x0040: 0000 0000 0006 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 001c
03:09:45.093144 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a b46b 0540 3131 .....k.@11
0x0020: 033d 10d2 0a0a 0001 0a0a 0001 b700   =.....
03:09:45.145128 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a dee2 0500 3000 .....0.
0x0020: 0334 10d2 0a0a 0004 0a0a 0002 8e00   4.....
03:09:45.226052 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd7 0500 3200 .....k..2.
0x0020: 0328 10d2 0a0a 000a 0a0a 000a ff00   (.
03:09:45.264863 IP 10.10.0.9.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a c698 0540 3100 .....@1.
0x0020: 0328 10d2 0a0a 000a 0a0a 000a a500   (.
03:09:45.270371 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a df99 0540 3100 .....@1.
0x0020: 0328 10d2 0a0a 000a 0a0a 000a 8c00   (.
03:09:45.294025 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a fedc 0500 3000 .....0.
0x0020: 0328 10d2 0a0a 000a 0a0a 0009 6e00   (.
03:09:45.310345 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd6 0500 3200 .....k..2.
0x0020: 032f 10d2 0a0a 0008 0a0a 0008 ff00   ./
03:09:47.081788 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p....
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea af40 0040 1176 83a9 fe00 05a3 758b D..@.@.v.....u.
0x0030: fd08 0000 3500 30a3 c600 0801 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3401 3002 3130 0231 3007 .....4.0.10.10.
0x0050: 696e
in
03:09:47.171465 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p....
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea b840 0040 1176 7aa9 fe00 05a3 758b D..@.@.vz.....u.
0x0030: fd08 0000 3500 30a3 c500 0901 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3401 3002 3130 0231 3007 .....4.0.10.10.
0x0050: 696e
in
03:09:47.201991 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p....
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea bb40 0040 1176 77a9 fe00 05a3 758b D..@.@.vw.....u.
0x0030: fd08 0000 3500 30a3 c400 0a01 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3401 3002 3130 0231 3007 .....4.0.10.10.
0x0050: 696e
in
03:09:47.210383 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a d1d1 0500 3100 .....1.
0x0020: 0336 10d2 0a0a 0004 0a0a 000a 9a00   6.....
03:09:47.216020 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p....
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea bc40 0040 1176 76a9 fe00 05a3 758b D..@.@.vv.....u.
0x0030: fd08 0000 3500 30a3 c100 0b01 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3601 3002 3130 0231 3007 .....6.0.10.10.
0x0050: 696e
in
03:09:47.216744 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 3f11 2770 0a0a 0008 E..a..@.?.p....
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea bc40 0040 1176 76a9 fe00 05a3 758b D..@.@.vv.....u.
0x0030: fd08 0000 3500 30a3 c100 0b01 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3601 3002 3130 0231 3007 .....6.0.10.10.

```

```

0x0050: 696e                                in
03:09:47.313462 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea c640 0040 1176 6ca9 fe00 05a3 758b D..@.@.vI....u.
0x0030: fd08 0000 3500 30a3 c000 0c01 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3601 3002 3130 0231 3007 .....6.0.10.10.
0x0050: 696e                                in
03:09:47.330523 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd4 0500 3200 .....k..2.
0x0020: 0331 10d2 0a0a 0008 0a0a 0008 ff00 ..1.....
03:09:47.334075 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea c840 0040 1176 6aa9 fe00 05a3 758b D..@.@.vj....u.
0x0030: fd08 0000 3500 30a3 bf00 0d01 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3601 3002 3130 0231 3007 .....6.0.10.10.
0x0050: 696e                                in
03:09:47.342411 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 70
0x0000: 4500 0062 0000 4000 4011 266f 0a0a 0008 E..b..@.@.&o....
0x0010: 0a0a 0001 10d2 10d2 004e 816b 0145 0000 .....N.k.E..
0x0020: 45ea c940 0040 1176 68a9 fe00 05a3 758b E..@.@.vh....u.
0x0030: fd08 0000 3500 310b 2900 0e01 0000 0100 ....5.1.).....
0x0040: 0000 0000 0002 3130 0130 0231 3002 3130 .....10.0.10.10
0x0050: 0769                                .i
03:09:47.435068 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 70
0x0000: 4500 0062 0000 4000 4011 266f 0a0a 0008 E..b..@.@.&o....
0x0010: 0a0a 0001 10d2 10d2 004e 816b 0145 0000 .....N.k.E..
0x0020: 45ea d240 0040 1176 5fa9 fe00 05a3 758b E..@.@.v....u.
0x0030: fd08 0000 3500 310b 2800 0f01 0000 0100 ....5.1.(.....
0x0040: 0000 0000 0002 3130 0130 0231 3002 3130 .....10.0.10.10
0x0050: 0769                                .i
03:09:47.459772 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 70
0x0000: 4500 0062 0000 4000 4011 266f 0a0a 0008 E..b..@.@.&o....
0x0010: 0a0a 0001 10d2 10d2 004e 816b 0145 0000 .....N.k.E..
0x0020: 45ea d440 0040 1176 5da9 fe00 05a3 758b E..@.@.vj....u.
0x0030: fd08 0000 3500 310b 2700 1001 0000 0100 ....5.1.'.....
0x0040: 0000 0000 0002 3130 0130 0231 3002 3130 .....10.0.10.10
0x0050: 0769                                .i
03:09:47.470608 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea d540 0040 1176 5da9 fe00 05a3 758b D..@.@.vI....u.
0x0030: fd08 0000 3500 30a3 be00 1101 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3301 3002 3130 0231 3007 .....3.0.10.10.
0x0050: 696e                                in
03:09:47.475807 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 3f11 2770 0a0a 0008 E..a..@.?.p....
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea d540 0040 1176 5da9 fe00 05a3 758b D..@.@.vI....u.
0x0030: fd08 0000 3500 30a3 be00 1101 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3301 3002 3130 0231 3007 .....3.0.10.10.
0x0050: 696e                                in
03:09:47.540354 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a d9d4 0500 3000 .....0.
0x0020: 033b 10d2 0a0a 0002 0a0a 0004 9300 ..;.....
03:09:47.622615 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea e540 0040 1176 4da9 fe00 05a3 758b D..@.@.vM....u.
0x0030: fd08 0000 3500 30a3 bd00 1201 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3301 3002 3130 0231 3007 .....3.0.10.10.
0x0050: 696e                                in
03:09:47.640619 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 25d0 0500 2e00 .....%.....
0x0020: 0339 10d2 0a0a 0003 0a0a 000a 4900 ..9.....I.
03:09:47.694392 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea ec40 0040 1176 46a9 fe00 05a3 758b D..@.@.vF....u.
0x0030: fd08 0000 3500 30a3 bc00 1301 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3301 3002 3130 0231 3007 .....3.0.10.10.
0x0050: 696e                                in

```

```

03:09:47.713651 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea ee40 0040 1176 44a9 fe00 05a3 758b D..@.@.vD.....u.
0x0030: fd08 0000 3500 30a3 bd00 1401 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3101 3002 3130 0231 3007 .....1.0.10.10.
0x0050: 696e
in

03:09:47.750367 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 28d1 0500 2f00 .....(../.
0x0020: 0335 10d2 0a0a 0006 0a0a 000a 4500 .....5.....E.

03:09:47.802151 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea f740 0040 1176 3ba9 fe00 05a3 758b D..@.@.v;.....u.
0x0030: fd08 0000 3500 30a3 bc00 1501 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3101 3002 3130 0231 3007 .....1.0.10.10.
0x0050: 696e
in

03:09:47.829860 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea f940 0040 1176 39a9 fe00 05a3 758b D..@.@.v9.....u.
0x0030: fd08 0000 3500 30a3 bb00 1601 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3101 3002 3130 0231 3007 .....1.0.10.10.
0x0050: 696e
in

03:09:47.885667 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea ff40 0040 1176 33a9 fe00 05a3 758b D..@.@.v3.....u.
0x0030: fd08 0000 3500 30a3 b900 1701 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3201 3002 3130 0231 3007 .....2.0.10.10.
0x0050: 696e
in

03:09:47.893919 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 3f11 2770 0a0a 0008 E..a..@.?.p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea ff40 0040 1176 33a9 fe00 05a3 758b D..@.@.v3.....u.
0x0030: fd08 0000 3500 30a3 b900 1701 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3201 3002 3130 0231 3007 .....2.0.10.10.
0x0050: 696e
in

03:09:47.894392 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 3e11 2870 0a0a 0008 E..a..@.>(p....
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44ea ff40 0040 1176 33a9 fe00 05a3 758b D..@.@.v3.....u.
0x0030: fd08 0000 3500 30a3 b900 1701 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3201 3002 3130 0231 3007 .....2.0.10.10.
0x0050: 696e
in

03:09:47.895389 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 116
0x0000: 4500 0090 0000 4000 4011 263f 0a0a 0001 E.....@.@.&?...
0x0010: 0a0a 000a 10d2 10d2 007c b10d 0145 0000 .....|.E..
0x0020: 7300 0040 003d 1164 02a3 758b fda9 fe00 s..@.=.d.u....
0x0030: 0700 3508 0000 5f8a 6000 0381 8000 0100 ..5...`.....
0x0040: 0000 0100 0006 676f 6f67 6c65 0265 7300 .....google.es.
0x0050: 001c
..

03:09:47.981555 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44eb 9400 0040 1176 29a9 fe00 05a3 758b D..@.@.v).....u.
0x0030: fd08 0000 3500 30a3 b800 1801 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3201 3002 3130 0231 3007 .....2.0.10.10.
0x0050: 696e
in

03:09:47.991671 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44eb 0a40 0040 1176 28a9 fe00 05a3 758b D..@.@.v(.....u.
0x0030: fd08 0000 3500 30a3 b700 1901 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3201 3002 3130 0231 3007 .....2.0.10.10.
0x0050: 696e
in

03:09:48.021904 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a c39c 0540 3100 .....@1.
0x0020: 0337 10d2 0a0a 0004 0a0a 0004 a800 .....7.....

03:09:48.054741 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a dadf 0500 3000 .....0.
0x0020: 0337 10d2 0a0a 0004 0a0a 0002 9200 .....7.....

03:09:48.109791 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69

```

```

0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44eb 1540 0040 1176 1da9 fe00 05a3 758b D..@.@.v.....u.
0x0030: fd08 0000 3500 30a3 af00 1a01 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3901 3002 3130 0231 3007 .....9.0.10.10.
0x0050: 696e                                     in
03:09:48.125026 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 3f11 2770 0a0a 0008 E..a..@.?.'p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44eb 1540 0040 1176 1da9 fe00 05a3 758b D..@.@.v.....u.
0x0030: fd08 0000 3500 30a3 af00 1a01 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3901 3002 3130 0231 3007 .....9.0.10.10.
0x0050: 696e                                     in
03:09:48.125739 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 3e11 2870 0a0a 0008 E..a..@.>.(p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44eb 1540 0040 1176 1da9 fe00 05a3 758b D..@.@.v.....u.
0x0030: fd08 0000 3500 30a3 af00 1a01 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3901 3002 3130 0231 3007 .....9.0.10.10.
0x0050: 696e                                     in
03:09:48.134749 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6ba9 0500 3231 .....k...21
0x0020: 0340 10d2 0a0a 0001 0a0a 0001 ff00      @.....
03:09:48.211671 IP 10.10.0.8.4306 > 10.10.0.1.4306: UDP, length 69
0x0000: 4500 0061 0000 4000 4011 2670 0a0a 0008 E..a..@.@.&p...
0x0010: 0a0a 0001 10d2 10d2 004d 806d 0145 0000 .....M.m.E..
0x0020: 44eb 2040 0040 1176 12a9 fe00 05a3 758b D..@.@.v.....u.
0x0030: fd08 0000 3500 30a3 ae00 1b01 0000 0100 ....5.0.....
0x0040: 0000 0000 0001 3901 3002 3130 0231 3007 .....9.0.10.10.
0x0050: 696e                                     in
03:09:48.267904 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd4 0500 3200 .....k...2.
0x0020: 032b 10d2 0a0a 000a 0a0a 000a ff00      .+.....
03:09:48.286309 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 279a 0500 2e31 .....!...1
0x0020: 0340 10d2 0a0a 0001 0a0a 0008 4700      @.....G.
03:09:48.375744 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd3 0500 3200 .....k...2.
0x0020: 0332 10d2 0a0a 0008 0a0a 0008 ff00      .2.....
03:09:48.376961 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a da96 0540 3100 .....@1.
0x0020: 032b 10d2 0a0a 000a 0a0a 000a 9100      .+.....
03:09:48.423605 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bdb 0500 3200 .....k...2.
0x0020: 033c 10d2 0a0a 0002 0a0a 0002 ff00      .<.....
03:09:48.431825 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 13dd 0500 3000 .....0.
0x0020: 032b 10d2 0a0a 000a 0a0a 0008 5900      .+.....Y.
03:09:48.433514 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 06de 0500 3000 .....0.
0x0020: 0332 10d2 0a0a 0008 0a0a 0004 6600      .2.....f.
03:09:48.451898 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a d9d6 0500 3000 .....0.
0x0020: 033c 10d2 0a0a 0002 0a0a 0006 9300      .<.....
03:09:48.463810 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 8e9c 0540 3100 .....@1.
0x0020: 033c 10d2 0a0a 0002 0a0a 0002 dd00      .<.....
03:09:48.464815 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 2be6 0500 2e00 .....+....
0x0020: 032b 10d2 0a0a 000a 0a0a 0002 4300      .+.....C.
03:09:48.486829 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 15e1 0500 2f00 ...../.
0x0020: 0332 10d2 0a0a 0008 0a0a 0002 5800      .2.....X.
03:09:48.526135 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18

```

```

0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 18cd 0500 2f00 ...../..
0x0020: 033c 10d2 0a0a 0002 0a0a 0009 5500 .....<.....U.
03:09:48.535521 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 22db 0500 2f00 ....."/..
0x0020: 0332 10d2 0a0a 0008 0a0a 0003 4b00 .....2.....K.
03:09:48.574873 IP 10.10.0.9.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd2 0500 3200 .....k..2.
0x0020: 0330 10d2 0a0a 0009 0a0a 0009 ff00 .....0.....
03:09:48.578344 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a b295 0540 3100 .....@1.
0x0020: 0330 10d2 0a0a 0009 0a0a 0009 b900 .....0.....
03:09:48.580653 IP 10.10.0.9.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a fed4 0500 3000 .....0.
0x0020: 0332 10d2 0a0a 0008 0a0a 0006 6e00 .....2.....n.
03:09:48.590364 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a e8cd 0500 3000 .....0.
0x0020: 033c 10d2 0a0a 0002 0a0a 000a 8400 .....<.....
03:09:48.591406 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a dedb 0500 3000 .....0.
0x0020: 0330 10d2 0a0a 0009 0a0a 0006 8e00 .....0.....
03:09:48.594220 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a bbd0 0500 3100 .....1.
0x0020: 0332 10d2 0a0a 0008 0a0a 0009 b000 .....2.....
03:09:48.614493 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a ffe0 0500 2f00 ...../..
0x0020: 0330 10d2 0a0a 0009 0a0a 0003 6e00 .....0.....n.
03:09:48.635520 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 20e2 0500 2e00 .....
0x0020: 0330 10d2 0a0a 0009 0a0a 0001 4e00 .....0.....N.
03:09:48.650148 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a b491 0540 3100 .....@1.
0x0020: 0330 10d2 0a0a 0009 0a0a 0009 b700 .....0.....
03:09:48.650205 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bda 0500 3200 .....k..2.
0x0020: 033a 10d2 0a0a 0003 0a0a 0003 ff00 .....:.....
03:09:48.674565 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a a59c 0540 3100 .....@1.
0x0020: 033a 10d2 0a0a 0003 0a0a 0003 c600 .....:.....
03:09:48.683043 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 1edf 0500 2f00 ...../..
0x0020: 0330 10d2 0a0a 0009 0a0a 0002 4f00 .....0.....O.
03:09:48.683869 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 08d3 0500 3000 .....0.
0x0020: 0330 10d2 0a0a 0009 0a0a 000a 6400 .....0.....d.
03:09:48.684187 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a ca97 0540 3100 .....@1.
0x0020: 033a 10d2 0a0a 0003 0a0a 0003 a100 .....:.....
03:09:48.706579 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd5 0500 3200 .....k..2.
0x0020: 0336 10d2 0a0a 0006 0a0a 0006 ff00 .....6.....
03:09:48.713049 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a dfdd 0500 3000 .....0.
0x0020: 033a 10d2 0a0a 0003 0a0a 0001 8d00 .....:.....
03:09:48.738381 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a ecda 0500 2f00 ...../..
0x0020: 033a 10d2 0a0a 0003 0a0a 0002 8100 .....:.....
03:09:48.750633 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18

```



```

0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 1ed5 0500 2e00 .....
0x0020: 033a 10d2 0a0a 0003 0a0a 0004 5000 .....P.
03:09:48.755177 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a a798 0540 3100 .....@1.
0x0020: 0336 10d2 0a0a 0006 0a0a 0006 c400 .....6.....
03:09:48.766514 IP 10.10.0.9.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a c792 0540 3100 .....@1.
0x0020: 0336 10d2 0a0a 0006 0a0a 0006 a400 .....6.....
03:09:48.787829 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 1ccf 0500 2f00 ...../.
0x0020: 033a 10d2 0a0a 0003 0a0a 0008 5100 .....Q.
03:09:48.818234 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a d4dd 0500 3000 .....0.
0x0020: 0336 10d2 0a0a 0006 0a0a 0003 9800 .....6.....
03:09:48.891804 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 0cdc 0500 2e00 .....
0x0020: 0336 10d2 0a0a 0006 0a0a 0002 6200 .....6.....b.
03:09:48.926401 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 4011 265e 0a0a 000a E..q..@.@.&^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@.@.....U.
0x0030: 6808 00ee 16d8 0200 01b5 8b7c 5900 0000 h.....|Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:09:48.926625 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3f11 275e 0a0a 000a E..q..@.?.'^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@.@.....U.
0x0030: 6808 00ee 16d8 0200 01b5 8b7c 5900 0000 h.....|Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:09:48.927448 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3e11 285e 0a0a 000a E..q..@.>.(^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@.@.....U.
0x0030: 6808 00ee 16d8 0200 01b5 8b7c 5900 0000 h.....|Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:09:48.927894 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3d11 295e 0a0a 000a E..q..@.(=).^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@.@.....U.
0x0030: 6808 00ee 16d8 0200 01b5 8b7c 5900 0000 h.....|Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:09:48.940359 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 27d6 0500 2f00 .....'./.
0x0020: 0336 10d2 0a0a 0006 0a0a 0004 4600 .....6.....F.
03:09:49.025445 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 4011 265e 0a0a 0001 E..q..@.@.&^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 545b 6200 0032 01cc 83d1 55e5 68a9 fe00 T[b..2....U.h...
0x0030: 0700 00f6 16d8 0200 01b5 8b7c 5900 0000 .....|Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:09:49.026769 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3f11 275e 0a0a 0001 E..q..@.?.'^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 545b 6200 0032 01cc 83d1 55e5 68a9 fe00 T[b..2....U.h...
0x0030: 0700 00f6 16d8 0200 01b5 8b7c 5900 0000 .....|Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:09:49.031344 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3e11 285e 0a0a 0001 E..q..@.>.(^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 545b 6200 0032 01cc 83d1 55e5 68a9 fe00 T[b..2....U.h...
0x0030: 0700 00f6 16d8 0200 01b5 8b7c 5900 0000 .....|Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..

```



```

03:09:49.033258 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
  0x0000: 4500 0071 0000 4000 3d11 295e 0a0a 0001 E...q..(.=.)^....
  0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
  0x0020: 545b 6200 0032 01cc 83d1 55e5 68a9 fe00 T[b..2....U.h...
  0x0030: 0700 00f6 16d8 0200 01b5 8b7c 5900 0000 .....]Y...
  0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
  0x0050: 0000 ..
03:09:49.084483 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a 6ba8 0500 3231 .....k..21
  0x0020: 0341 10d2 0a0a 0001 0a0a 0001 ff00 ..A.....
03:09:49.226092 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a 6bd3 0500 3200 .....k..2.
  0x0020: 032c 10d2 0a0a 000a 0a0a 000a ff00 .....
03:09:49.256492 IP 10.10.0.9.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a c894 0540 3100 .....@1.
  0x0020: 032c 10d2 0a0a 000a 0a0a 000a a300 .....
03:09:49.280358 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a df95 0540 3100 .....@1.
  0x0020: 032c 10d2 0a0a 000a 0a0a 000a 8c00 .....
03:09:49.340338 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a 6bd2 0500 3200 .....k..2.
  0x0020: 0333 10d2 0a0a 0008 0a0a 0008 ff00 ..3.....
03:09:49.391912 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a 39e2 0500 2f00 .....9.../.
  0x0020: 032c 10d2 0a0a 000a 0a0a 0004 3400 .....4.
03:09:49.486245 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a 1be4 0500 2f00 ...../.
  0x0020: 032c 10d2 0a0a 000a 0a0a 0001 5200 .....R.
03:09:49.514960 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a ffde 0500 3000 .....0.
  0x0020: 032c 10d2 0a0a 000a 0a0a 0003 6d00 .....m.
03:09:49.529416 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a 6bda 0500 3200 .....k..2.
  0x0020: 033d 10d2 0a0a 0002 0a0a 0002 ff00 ..=.....
03:09:49.530334 IP 10.10.0.9.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a 6bd1 0500 3200 .....k..2.
  0x0020: 0331 10d2 0a0a 0009 0a0a 0009 ff00 ..1.....
03:09:49.544967 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a 0e97 0540 2f00 .....@/..
  0x0020: 033d 10d2 0a0a 0002 0a0a 0002 5f00 ..=.....
03:09:49.561699 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a d8d5 0500 3000 .....0.
  0x0020: 033d 10d2 0a0a 0002 0a0a 0006 9400 ..=.....
03:09:49.583792 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a 8598 0540 3100 .....@1.
  0x0020: 033d 10d2 0a0a 0002 0a0a 0002 e600 ..=.....
03:09:49.605817 IP 10.10.0.9.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a 32d0 0500 2e00 .....2....
  0x0020: 033d 10d2 0a0a 0002 0a0a 0006 3c00 ..=.....<.
03:09:49.660348 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a e0d2 0500 3000 .....0.
  0x0020: 033d 10d2 0a0a 0002 0a0a 0004 8c00 ..=.....
03:09:49.676148 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a 0ecd 0500 2f00 ...../.
  0x0020: 033d 10d2 0a0a 0002 0a0a 0008 5f00 ..=.....
03:09:49.689878 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
  0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E....@.@.&....
  0x0010: 0a0a ffff 10d1 10d1 001a 6bd9 0500 3200 .....k..2.
  0x0020: 033b 10d2 0a0a 0003 0a0a 0003 ff00 .....
03:09:49.714161 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18

```

```

0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a ca96 0540 3100 .....@1.
0x0020: 033b 10d2 0a0a 0003 0a0a 0003 a100 .....;.....
03:09:49.741856 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a e3dc 0500 3000 .....0.
0x0020: 033b 10d2 0a0a 0003 0a0a 0001 8900 .....;.....
03:09:49.792004 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a ac97 0540 3100 .....@1.
0x0020: 0337 10d2 0a0a 0006 0a0a 0006 bf00 .....7.....
03:09:49.803912 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a f2d9 0500 2f00 ...../.
0x0020: 033b 10d2 0a0a 0003 0a0a 0002 7b00 .....;.....{.
03:09:49.808732 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 24cd 0500 2f00 .....$../.
0x0020: 033b 10d2 0a0a 0003 0a0a 0009 4900 .....;.....I.
03:09:49.830353 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 24d4 0500 2e00 .....$.
0x0020: 033b 10d2 0a0a 0003 0a0a 0004 4a00 .....;.....J.
03:09:49.871961 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 07de 0500 2f00 ...../.
0x0020: 0337 10d2 0a0a 0006 0a0a 0001 6600 .....7.....f.
03:09:49.911427 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 12db 0500 2e00 .....
0x0020: 0337 10d2 0a0a 0006 0a0a 0002 5c00 .....7.....\
03:09:49.927938 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 4011 265e 0a0a 000a E..q..@.@.&^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@.@.....U.
0x0030: 6808 0099 d3d8 0200 02fa cd8b 5900 0000 h.....Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:09:49.928163 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3f11 275e 0a0a 000a E..q..@.?.^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@.@.....U.
0x0030: 6808 0099 d3d8 0200 02fa cd8b 5900 0000 h.....Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:09:49.930218 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3e11 285e 0a0a 000a E..q..@.>.(^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@.@.....U.
0x0030: 6808 0099 d3d8 0200 02fa cd8b 5900 0000 h.....Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:09:49.930726 IP 10.10.0.10.4306 > 10.10.0.1.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3d11 295e 0a0a 000a E..q..@.=.)^....
0x0010: 0a0a 0001 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 5400 0040 0040 01d9 e5a9 fe00 07d1 55e5 T..@.@.....U.
0x0030: 6808 0099 d3d8 0200 02fa cd8b 5900 0000 h.....Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:09:49.951504 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd8 0500 3200 .....k..2.
0x0020: 0339 10d2 0a0a 0004 0a0a 0004 ff00 .....9.....
03:09:49.956029 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 4011 265e 0a0a 0001 E..q..@.@.&^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 545b 6300 0032 01cc 82d1 55e5 68a9 fe00 T[c..2....U.h...
0x0030: 0700 00a1 d3d8 0200 02fa cd8b 5900 0000 .....Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:09:49.957565 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3f11 275e 0a0a 0001 E..q..@.?.^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 545b 6300 0032 01cc 82d1 55e5 68a9 fe00 T[c..2....U.h...
0x0030: 0700 00a1 d3d8 0200 02fa cd8b 5900 0000 .....Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....

```

```

0x0050: 0000
03:09:49.970899 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 28d5 0500 2f00 .....(/.
0x0020: 0337 10d2 0a0a 0006 0a0a 0004 4500 .7.....E.
03:09:49.974278 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 2997 0540 2e00 .....).@..
0x0020: 0339 10d2 0a0a 0004 0a0a 0004 4500 .9.....E.
03:09:49.982091 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a ca9a 0540 3100 .....@1.
0x0020: 0339 10d2 0a0a 0004 0a0a 0004 a100 .9.....
03:09:49.998744 IP 10.10.0.9.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 2ed2 0500 2f00 ...../.
0x0020: 0339 10d2 0a0a 0004 0a0a 0006 3f00 .9.....?.
03:09:50.000368 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a cb94 0540 3100 .....@1.
0x0020: 0339 10d2 0a0a 0004 0a0a 0004 a000 .9.....
03:09:50.007038 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3e11 285e 0a0a 0001 E..q.@.>.(^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 545b 6300 0032 01cc 82d1 55e5 68a9 fe00 T[c..2...U.h...
0x0030: 0700 00a1 d3d8 0200 02fa cd8b 5900 0000 .....Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000
03:09:50.012959 IP 10.10.0.1.4306 > 10.10.0.10.4306: UDP, length 85
0x0000: 4500 0071 0000 4000 3d11 295e 0a0a 0001 E..q.@.=.)^....
0x0010: 0a0a 000a 10d2 10d2 005d c871 0145 0000 .....].q.E..
0x0020: 545b 6300 0032 01cc 82d1 55e5 68a9 fe00 T[c..2...U.h...
0x0030: 0700 00a1 d3d8 0200 02fa cd8b 5900 0000 .....Y...
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000
03:09:50.031182 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 0dd8 0500 2f00 ...../.
0x0020: 0339 10d2 0a0a 0004 0a0a 0006 6000 .9.....`.
03:09:50.077442 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a dbdc 0500 3000 .....0.
0x0020: 0339 10d2 0a0a 0004 0a0a 0003 9100 .9.....
03:09:50.101875 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a b766 0540 3131 .....f.@11
0x0020: 0342 10d2 0a0a 0001 0a0a 0001 b400 .B.....
03:09:50.161407 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a caa3 0500 3031 .....01
0x0020: 0342 10d2 0a0a 0001 0a0a 0002 a200 .B.....
03:09:50.175153 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd2 0500 3200 .....k..2.
0x0020: 032d 10d2 0a0a 000a 0a0a 000a ff00 .-.....
03:09:50.176046 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 089e 0500 2f31 ...../1
0x0020: 0342 10d2 0a0a 0001 0a0a 0004 6500 .B.....e.
03:09:50.200348 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a e794 0540 3100 .....@1.
0x0020: 032d 10d2 0a0a 000a 0a0a 000a 8400 .-.....K.
03:09:50.221225 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 2a98 0500 2e31 .....*...1
0x0020: 0342 10d2 0a0a 0001 0a0a 0008 4400 .B.....D.
03:09:50.223323 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 21db 0500 3000 .....!...0.
0x0020: 032d 10d2 0a0a 000a 0a0a 0008 4b00 .-.....K.
03:09:50.235095 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E.....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a fad7 0500 3000 .....0.
0x0020: 032d 10d2 0a0a 000a 0a0a 0009 7200 .-.....r.
03:09:50.236012 IP 10.10.0.9.4305 > 10.10.255.255.4305: UDP, length 18

```

```

0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 2397 0500 2f31 .....#.../1
0x0020: 0342 10d2 0a0a 0001 0a0a 000a 4a00 .....B.....J.
03:09:50.264083 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a fa9a 0500 3031 .....01
0x0020: 0342 10d2 0a0a 0001 0a0a 0009 7200 .....B.....r.
03:09:50.280149 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 3ee1 0500 2f00 .....>.../.
0x0020: 032d 10d2 0a0a 000a 0a0a 0004 2f00 .....-...../.
03:09:50.291137 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 18de 0500 2f00 ...../..
0x0020: 032d 10d2 0a0a 000a 0a0a 0006 5500 .....-.....U.
03:09:50.296790 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 2be4 0500 2e00 .....+.....
0x0020: 032d 10d2 0a0a 000a 0a0a 0002 4300 .....-.....C.
03:09:50.312560 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a c0a0 0500 3131 .....11
0x0020: 0342 10d2 0a0a 0001 0a0a 0006 ab00 .....B.....
03:09:50.390348 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd1 0500 3200 .....k...2.
0x0020: 0334 10d2 0a0a 0008 0a0a 0008 ff00 .....4.....
03:09:50.411525 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a d595 0540 3100 .....@1.
0x0020: 0334 10d2 0a0a 0008 0a0a 0008 9600 .....4.....
03:09:50.427991 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a b58f 0540 3100 .....@1.
0x0020: 0334 10d2 0a0a 0008 0a0a 0008 b600 .....4.....
03:09:50.462054 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 0ddc 0500 3000 .....0.
0x0020: 0334 10d2 0a0a 0008 0a0a 0004 5f00 .....4.....
03:09:50.484727 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 1bdf 0500 2f00 ...../..
0x0020: 0334 10d2 0a0a 0008 0a0a 0002 5200 .....4.....R.
03:09:50.531149 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 36de 0500 2e00 .....6....
0x0020: 0334 10d2 0a0a 0008 0a0a 0001 3800 .....4.....8.
03:09:50.564321 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 21d9 0500 2f00 .....!.../.
0x0020: 0334 10d2 0a0a 0008 0a0a 0003 4c00 .....4.....L.
03:09:50.586021 IP 10.10.0.9.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a fdd2 0500 3000 .....0.
0x0020: 0334 10d2 0a0a 0008 0a0a 0006 6f00 .....4.....o.
03:09:50.609794 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd9 0500 3200 .....k...2.
0x0020: 033e 10d2 0a0a 0002 0a0a 0002 ff00 .....>.....
03:09:50.628964 IP 10.10.0.9.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd0 0500 3200 .....k...2.
0x0020: 0332 10d2 0a0a 0009 0a0a 0009 ff00 .....2.....
03:09:50.644107 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 0e96 0540 2f00 .....@/..
0x0020: 033e 10d2 0a0a 0002 0a0a 0002 5f00 .....>.....
03:09:50.646469 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a b48f 0540 3100 .....@1.
0x0020: 0332 10d2 0a0a 0009 0a0a 0009 b700 .....2.....
03:09:50.647436 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd3 0500 3200 .....k...2.
0x0020: 0338 10d2 0a0a 0006 0a0a 0006 ff00 .....8.....
03:09:50.660145 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18

```

```

0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 8597 0540 3100 .....@1.
0x0020: 033e 10d2 0a0a 0002 0a0a 0002 e600 >.....
03:09:50.665104 IP 10.10.0.9.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 33cf 0500 2e00 .....3....
0x0020: 033e 10d2 0a0a 0002 0a0a 0006 3b00 >.....;
03:09:50.681229 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a ac96 0540 3100 .....@1.
0x0020: 0338 10d2 0a0a 0006 0a0a 0006 bf00 .8.....
03:09:50.683650 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a af93 0540 3100 .....@1.
0x0020: 0332 10d2 0a0a 0009 0a0a 0009 bc00 .2.....
03:09:50.695357 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a d8db 0500 3000 .....0.
0x0020: 0338 10d2 0a0a 0006 0a0a 0003 9400 .8.....
03:09:50.696267 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a f28f 0540 3000 .....@0.
0x0020: 0338 10d2 0a0a 0006 0a0a 0006 7a00 .8.....z.
03:09:50.697167 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a d7d1 0500 3000 .....0.
0x0020: 033e 10d2 0a0a 0002 0a0a 0004 9500 >.....
03:09:50.713864 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 2ece 0500 2f00 ...../.
0x0020: 0338 10d2 0a0a 0006 0a0a 000a 3f00 .8.....?.
03:09:50.731998 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd8 0500 3200 .....k..2.
0x0020: 033c 10d2 0a0a 0003 0a0a 0003 ff00 .<.....
03:09:50.732893 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a dfd9 0500 3000 .....0.
0x0020: 0332 10d2 0a0a 0009 0a0a 0006 8d00 .2.....
03:09:50.740439 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 0dd1 0500 3000 .....0.
0x0020: 0332 10d2 0a0a 0009 0a0a 000a 5f00 .2.....
03:09:50.742606 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 05dd 0500 2f00 ...../.
0x0020: 0338 10d2 0a0a 0006 0a0a 0001 6800 .8.....h.
03:09:50.761678 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 0ed4 0500 2e00 .....
0x0020: 0338 10d2 0a0a 0006 0a0a 0008 6000 .8.....`.
03:09:50.762588 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 36d7 0500 2f00 .....6.../.
0x0020: 0332 10d2 0a0a 0009 0a0a 0008 3700 .2.....7.
03:09:50.763565 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a c595 0540 3100 .....@1.
0x0020: 033c 10d2 0a0a 0003 0a0a 0003 a600 .<.....
03:09:50.787653 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 00df 0500 2f00 ...../.
0x0020: 0332 10d2 0a0a 0009 0a0a 0003 6d00 .2.....m.
03:09:50.814608 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 29dd 0500 2e00 .....).
0x0020: 0332 10d2 0a0a 0009 0a0a 0004 4500 .2.....E.
03:09:50.825080 IP 10.10.0.9.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 02d0 0500 3000 .....0.
0x0020: 033c 10d2 0a0a 0003 0a0a 0006 6a00 .<.....j.
03:09:50.846235 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 22cc 0500 2f00 ....."/.
0x0020: 033c 10d2 0a0a 0003 0a0a 0009 4b00 .<.....K.
03:09:50.860602 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18

```

```

0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 22cd 0500 2e00 .....".
0x0020: 033c 10d2 0a0a 0003 0a0a 000a 4c00 .....<.....L.
03:09:50.913450 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a f3d2 0500 2f00 ...../.
0x0020: 033c 10d2 0a0a 0003 0a0a 0008 7a00 .....<.....z.
03:09:50.961559 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd7 0500 3200 .....k..2.
0x0020: 033a 10d2 0a0a 0004 0a0a 0004 ff00 .....:.....
03:09:51.010347 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a c593 0540 3100 .....@1.
0x0020: 033a 10d2 0a0a 0004 0a0a 0004 a600 .....:.....
03:09:51.021893 IP 10.10.0.2.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26aa 0a0a 0002 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a ca99 0540 3100 .....@1.
0x0020: 033a 10d2 0a0a 0004 0a0a 0004 a100 .....:.....
03:09:51.028175 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a f6cd 0500 3000 .....0.
0x0020: 033a 10d2 0a0a 0004 0a0a 0008 7600 .....:.....v.
03:09:51.064920 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a e0dc 0500 3000 .....0.
0x0020: 033a 10d2 0a0a 0004 0a0a 0002 8c00 .....:.....
03:09:51.174773 IP 10.10.0.1.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26ab 0a0a 0001 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6ba6 0500 3231 .....k..21
0x0020: 0343 10d2 0a0a 0001 0a0a 0001 ff00 .....C.....
03:09:51.192920 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a bc64 0540 3131 .....d..@11
0x0020: 0343 10d2 0a0a 0001 0a0a 0001 af00 .....C.....
03:09:51.217906 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a fa9f 0500 3031 .....01
0x0020: 0343 10d2 0a0a 0001 0a0a 0003 7200 .....C.....r.
03:09:51.224022 IP 10.10.0.4.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a8 0a0a 0004 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a caa2 0500 3031 .....01
0x0020: 0343 10d2 0a0a 0001 0a0a 0002 a200 .....C.....
03:09:51.275540 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 6bd1 0500 3200 .....k..2.
0x0020: 032e 10d2 0a0a 000a 0a0a 000a ff00 .....:.....
03:09:51.276429 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 049d 0500 2f31 ...../1
0x0020: 0343 10d2 0a0a 0001 0a0a 0004 6900 .....C.....i.
03:09:51.290600 IP 10.10.0.8.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a4 0a0a 0008 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a e793 0540 3100 .....@1.
0x0020: 032e 10d2 0a0a 000a 0a0a 000a 8400 .....:.....
03:09:51.304944 IP 10.10.0.9.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a c892 0540 3100 .....@1.
0x0020: 032e 10d2 0a0a 000a 0a0a 000a a300 .....:.....
03:09:51.305864 IP 10.10.0.10.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a2 0a0a 000a E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 2697 0500 2e31 .....&...1
0x0020: 0343 10d2 0a0a 0001 0a0a 0008 4800 .....C.....H.
03:09:51.326776 IP 10.10.0.9.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a3 0a0a 0009 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 2396 0500 2f31 .....#.../1
0x0020: 0343 10d2 0a0a 0001 0a0a 000a 4a00 .....C.....J.
03:09:51.364459 IP 10.10.0.6.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a6 0a0a 0006 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a f7d6 0500 3000 .....0.
0x0020: 032e 10d2 0a0a 000a 0a0a 0009 7500 .....:.....u.
03:09:51.411169 IP 10.10.0.3.4305 > 10.10.255.255.4305: UDP, length 18
0x0000: 4500 002e 0000 4000 4011 26a9 0a0a 0003 E....@.@.&....
0x0010: 0a0a ffff 10d1 10d1 001a 13dd 0500 2f00 ...../.
0x0020: 032e 10d2 0a0a 000a 0a0a 0006 5a00 .....:.....Z.

```

root@OpenWrt:~#



## A2.10. CAPTURA DE TRÁFICO FUERA DE LA RED MALLADA

Nos encontramos en la misma situación que se muestra en la Figura 4.9 y vamos a mostrar la captura de tráfico fuera de la red mallada inalámbrica. Para ello capturamos en el nodo gateway en la interfaz eth0.1 que es la que proporciona el acceso a Internet. El comando que hay que ejecutar es el siguiente:

```
tcpdump -i eth0.1 -X > fuera
```

El fichero “fuera” es el que contiene el resultado de la captura y el que mostraremos a continuación. Las IPs utilizadas son:

**163.117.140.216** → eth0.1 → **a375 8cd8**  
**163.117.139.253** → DNS **varpa.it.uc3m.es** → **a375 8bfd**  
**209.85.229.104** → google.es / **ww-in-f104.1e100.net** → **d155 e568**

```
root@OpenWrt:~# cat fuera
03:37:36.692447 IP 163.117.140.216.2048 > varpa.it.uc3m.es. 53: 2+ AAAA? google.es. (27)
0x0000: 4500 0037 5212 4000 3f11 89e3 a375 8cd8 E..7R.@.?.....u..
0x0010: a375 8bfd 0800 0035 0023 d1f6 0002 0100 .u....5#.....
0x0020: 0001 0000 0000 0000 0667 6f6f 676c 6502 .....google.
0x0030: 6573 0000 1c00 01 es.....
03:37:36.693070 IP varpa.it.uc3m.es. 53 > 163.117.140.216.2048: 2 0/1/0 (87)
0x0000: 4500 0073 0000 4000 3e11 dcb9 a375 8bfd E..s..@.>.....u..
0x0010: a375 8cd8 0035 0800 005f fe16 0002 8180 .u...5..._.....
0x0020: 0001 0000 0001 0000 0667 6f6f 676c 6502 .....google.
0x0030: 6573 0000 1c00 01c0 0c00 0600 0100 0000 es.....
0x0040: 3000 3003 6e73 3106 676f 6f67 6c65 0363 0.0.ns1.google.c
0x0050: 6f6d om
03:37:36.712077 IP 163.117.140.216.2048 > varpa.it.uc3m.es. 53: 3+ AAAA? google.es. (27)
0x0000: 4500 0037 5213 4000 3f11 89e2 a375 8cd8 E..7R.@.?.....u..
0x0010: a375 8bfd 0800 0035 0023 d1f5 0003 0100 .u....5#.....
0x0020: 0001 0000 0000 0000 0667 6f6f 676c 6502 .....google.
0x0030: 6573 0000 1c00 01 es.....
03:37:36.712585 IP varpa.it.uc3m.es. 53 > 163.117.140.216.2048: 3 0/1/0 (87)
0x0000: 4500 0073 0000 4000 3e11 dcb9 a375 8bfd E..s..@.>.....u..
0x0010: a375 8cd8 0035 0800 005f fe15 0003 8180 .u...5..._.....
0x0020: 0001 0000 0001 0000 0667 6f6f 676c 6502 .....google.
0x0030: 6573 0000 1c00 01c0 0c00 0600 0100 0000 es.....
0x0040: 3000 3003 6e73 3106 676f 6f67 6c65 0363 0.0.ns1.google.c
0x0050: 6f6d om
03:37:36.724164 IP 163.117.140.216.2048 > varpa.it.uc3m.es. 53: 4+ AAAA? google.es. (27)
0x0000: 4500 0037 5215 4000 3f11 89e0 a375 8cd8 E..7R.@.?.....u..
0x0010: a375 8bfd 0800 0035 0023 d1f4 0004 0100 .u....5#.....
0x0020: 0001 0000 0000 0000 0667 6f6f 676c 6502 .....google.
0x0030: 6573 0000 1c00 01 es.....
03:37:36.724706 IP varpa.it.uc3m.es. 53 > 163.117.140.216.2048: 4 0/1/0 (87)
0x0000: 4500 0073 0000 4000 3e11 dcb9 a375 8bfd E..s..@.>.....u..
0x0010: a375 8cd8 0035 0800 005f fe14 0004 8180 .u...5..._.....
0x0020: 0001 0000 0001 0000 0667 6f6f 676c 6502 .....google.
0x0030: 6573 0000 1c00 01c0 0c00 0600 0100 0000 es.....
0x0040: 3000 3003 6e73 3106 676f 6f67 6c65 0363 0.0.ns1.google.c
0x0050: 6f6d om
03:37:36.737958 IP 163.117.140.216.2048 > varpa.it.uc3m.es. 53: 5+ A? google.es. (27)
0x0000: 4500 0037 5217 4000 3f11 89de a375 8cd8 E..7R.@.?.....u..
0x0010: a375 8bfd 0800 0035 0023 ecf3 0005 0100 .u....5#.....
0x0020: 0001 0000 0000 0000 0667 6f6f 676c 6502 .....google.
0x0030: 6573 0000 0100 01 es.....
03:37:36.738638 IP varpa.it.uc3m.es. 53 > 163.117.140.216.2048: 5 3/4/4 A ww-in-f104.1e100.net,[domain]
0x0000: 4500 00f9 0000 4000 3e11 dc33 a375 8bfd E.....@.>...3.u..
0x0010: a375 8cd8 0035 0800 00e5 42fb 0005 8180 .u...5...B....
0x0020: 0001 0003 0004 0004 0667 6f6f 676c 6502 .....google.
0x0030: 6573 0000 0100 01c0 0c00 0100 0100 0005 es.....
0x0040: c700 04d1 55e5 68c0 0c00 0100 0100 0005 ....U.h.....
0x0050: c700 ..
03:37:36.820325 IP 163.117.140.216 >: ICMP echo request, id 57346, seq 0, length 64
```

```

0x0000: 4500 0054 0000 4000 3f01 549d a375 8cd8 E..T..@.?.T..u..
0x0010: d155 e568 0800 3bde e002 0000 8266 59b8 .U.h.,.....fY.
0x0020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:37:36.843978 IP ww-in-f104.1e100.net > 163.117.140.216: ICMP echo reply, id 57346, seq 0, length 64
0x0000: 4500 0054 7320 0000 3201 2e7d d155 e568 E..Ts...2..}.U.h
0x0010: a375 8cd8 0000 43de e002 0000 8266 59b8 .u...C.....fY.
0x0020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:37:37.346833 STP 802.1d, Config, Flags [none], bridge-id 8000.00:15:c5:de:88:05.8002, length 43
0x0000: 4242 0300 0000 0000 2000 0019 a92f ec06 BB...../..
0x0010: 0000 001b 8000 0015 c5de 8805 8002 0300 .....
0x0020: 1400 0200 0f00 0000 0000 0000 0000 0000 .....
03:37:37.761616 IP 163.117.140.216 > ww-in-f104.1e100.net: ICMP echo request, id 57346, seq 1, length 64
0x0000: 4500 0054 0000 4000 3f01 549d a375 8cd8 E..T..@.?.T..u..
0x0010: d155 e568 0800 c8a3 e002 0001 e69f 68b8 .U.h.....h.
0x0020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:37:37.785564 IP ww-in-f104.1e100.net > 163.117.140.216: ICMP echo reply, id 57346, seq 1, length 64
0x0000: 4500 0054 7321 0000 3201 2e7c d155 e568 E..Ts!..2..|.U.h
0x0010: a375 8cd8 0000 d0a3 e002 0001 e69f 68b8 .u.....h.
0x0020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:37:38.820854 IP 163.117.140.216 > ww-in-f104.1e100.net: ICMP echo request, id 57346, seq 2, length 64
0x0000: 4500 0054 0000 4000 3f01 549d a375 8cd8 E..T..@.?.T..u..
0x0010: d155 e568 0800 5860 e002 0002 47e2 77b8 .U.h..X'....G.w.
0x0020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
03:37:38.844319 IP ww-in-f104.1e100.net > 163.117.140.216: ICMP echo reply, id 57346, seq 2, length 64
0x0000: 4500 0054 7322 0000 3201 2e7b d155 e568 E..Ts"...2..{.U.h
0x0010: a375 8cd8 0000 6060 e002 0002 47e2 77b8 .u....``....G.w.
0x0020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0050: 0000 ..
root@OpenWrt:~#

```



# Referencias.

- [1] Ian F. Akyildiz, Xudong Wang, Weilin Wang: "[Wireless mesh networks: a survey](#)", "Computer Networks". Vol. 47. No.4 (2005) Pág. 455-487.
- [2] Diccionario de informática. <http://www.alegsa.com.ar/Dic/inalambrico.php> [Consulta: Marzo 2010].
- [3] Sebastián Büttrich. less.dk wire. "Unidad 13: Mesh". TRICALCAR. Versión final. (Octubre 2007) Pág. 4-5.  
[http://www.eslared.org.ve/tricalcar/13\\_es\\_redes\\_mesh\\_guia\\_v02%5B1%5D.pdf](http://www.eslared.org.ve/tricalcar/13_es_redes_mesh_guia_v02%5B1%5D.pdf) [Consulta: Diciembre 2009].
- [4] Diana Margarita Acuña Martínez, Rafael Julio Roncallo Kelsey: REDES INALÁMBRICAS ENMALLADAS METROPOLITANAS. (Octubre 2006) Pág. 46-91.  
<http://www.monografias.com/trabajos-pdf/redes-inalambricas-enmalladas-metropolitanas/redes-inalambricas-enmalladas-metropolitanas.pdf> [Consulta: Diciembre 2009].
- [5] Kolbjorn Tunstrom, "Simulation of B.A.T.M.A.N." (Enero 2008) Pág. 5  
[http://www.olsr.org/~aaron/olsr/batman\\_report.pdf](http://www.olsr.org/~aaron/olsr/batman_report.pdf) [Consulta: Junio 2010]
- [6] Mónica Pérez de Larraya: "La red mallada, un paso más allá en las Ciudades Digitales". Revista Auditoría y Seguridad. Número 21, Abril 2008.  
<http://www.revista-ays.com/DocsNum21/PersEmpresarial/Perez.pdf>
- [7] TUREGANO.NET: "Enredando con el Mesh"  
<http://www.turegano.net/2007/09/21/enredando-con-el-mesh/> [Consulta: Abril 2010].
- [8] Baldomero Coll, Javier Gozávez y Jonathan Rodríguez: "Técnicas Multi-Hop para Uso Eficiente de los Recursos en Redes de Comunicación 4G". Telecom I+D, 2009.  
[http://www.uwicare.umh.es/files/paper/2009\\_national/uwicare TELECOM\\_Tecnicas%20Multihop%20para%20uso%20eficiente%20de%20recursos%20en%20Redes%20de%20Comunicacion%204G.pdf](http://www.uwicare.umh.es/files/paper/2009_national/uwicare TELECOM_Tecnicas%20Multihop%20para%20uso%20eficiente%20de%20recursos%20en%20Redes%20de%20Comunicacion%204G.pdf)
- [9] P. Gupta, P.R. Kumar, The capacity of wireless networks, IEEE Transactions on Information Theory 46 (2) (2000) Pág. 388–404.
- [10] Brad Karp and H. T. Kung. GPRS: "Greedy perimeter stateless routing for wireless networks". In ACM/IEEE International Conference on Mobile Computing and Networking Mobicom, 2000.
- [11] Reference Manual of B.A.T.M.A.N. Experimental  
<http://downloads.open-mesh.net/batman/misc/bmx.pdf> [Consulta: Octubre 2009]

- [12] Curso de Redes inalámbricas WIFI seguras  
<http://www.adrformacion.com/cursos/wifi/leccion2/tutorial3.html>
- [13] Roberto Subiela Dura y Antonio León Fernández. Universidad Politécnica de Valencia. “Encaminamiento en redes móviles ad hoc con NS-2”.  
[http://w3.iec.csic.es/URSI/articulos\\_modernos/articulos\\_gandia\\_2005/articulos/TE2/396.pdf](http://w3.iec.csic.es/URSI/articulos_modernos/articulos_gandia_2005/articulos/TE2/396.pdf)
- [14] Algoritmos Cross-Layer para la Optimización de las prestaciones del TCP en Redes Wireless Ad-hoc Capítulo 3. “REDES WIRELESS AD-HOC”. (2005-2006) Pág. 49-50  
<http://bibing.us.es/proyectos/abreproy/11306/fichero/TEORIA%252F08+-+Capitulo+3.pdf>
- [15] A. Neumann, C. Aichele, M. Linder y S. Wunderlich. Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.) IETF Trust. draft-wunderlich-openmesh-manet.routing-00. (Abril 2008)  
<http://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00>
- [16] Modos de funcionamiento Wifi (802.11 o Wi-Fi).  
<http://es.kioskea.net/contents/wifi/wifimodes.php3>  
[Consulta: Octubre 2009]
- [17] Axel Neumann, Corinna “Elecktra” Aichele, Marek Lindner. B.A.T.M.A.N. Status Report, June 28, 2007. Pág. 2-3  
<http://downloads.open-mesh.net/batman/misc/batman-status.pdf>  
[Consulta: Octubre 2009]
- [18] Axel Neumann, [B.A.T.M.A.N.] one-way / two-way tunnel (was: Memory leak)  
<https://lists.open-mesh.org/pipermail/b.a.t.m.a.n/2008-January/000517.html>  
[Consulta: Octubre 2009]
- [19] TCP/IP Port Numbers (TCP and UDP) – Network Services (IANA)  
<http://www.honeypots.net/misc/services/>  
[Consulta: Mayo 2010]
- [20] OpenWrt en mi router Linksys  
<http://www.miltonpivori.com.ar/2007/10/16/openwrt-en-mi-router-linksys/>  
[Consulta: Octubre 2009]
- [21] OpenWrt en mi Linksys WRT 54G  
<http://www.modemhead.com.ar/2008/09/openwrt-en-mi-linksys-wrt-54g/>  
[Consulta: Octubre 2009]
- [22] Sacale Provecho a tu Router - OpenWRT  
<http://www.taringa.net/posts/info/1045753/Sacale-Provecho-a-tu-Router---OpenWRT.html>  
[Consulta: Octubre 2009]
- [23] Installing OpenWRT and OpenSWAN

- <http://mikemcarthur.net/article.php?story=20050613041748358>  
[Consulta: Octubre 2009]
- [24] Iptables mac address filtering  
<http://www.cyberciti.biz/tips/iptables-mac-address-filtering.html>  
[Consulta: Enero 2010]
- [25] Recuperar mediante tftp un WRT54G/S de una carga incorrecta de firmware  
<http://www.badalonawireless.net/node/246>  
[Consulta: Julio 2009]
- [26] Why Public Key Authentication  
<http://wiki.openwrt.org/oldwiki/dropbearpublickeyauthenticationhowto>  
[Consulta: Febrero 2010]
- [27] fping <http://www.mksoftware.com/docs/man1/fping.1.asp>  
[Consulta: Noviembre 2009]
- [28] tcpdump <http://es.wikipedia.org/wiki/Tcpdump>  
[Consulta: Noviembre 2009]
- [29] Saulo Barajas. Curso de protocolos TCP/IP. 2.5. Formato del datagrama IP  
<http://www.saulo.net/pub/tcpip/a.htm>  
[Consulta: Marzo 2010]
- [30] User Datagram Protocol [http://es.wikipedia.org/wiki/User\\_Datagram\\_Protocol](http://es.wikipedia.org/wiki/User_Datagram_Protocol)  
[Consulta: Marzo 2010]
- [31] Internet tunnig <http://www.open-mesh.net/wiki/InternetTuning>  
[Consulta: Diciembre 2009]